Aviation Human Factors Division
Institute of Aviation

University of Illinois
at Urbana-Champaign
1 Airport Road
Savoy, Illinois 61874

Ecological Validity as a Mediator of Visual Attention Allocation in Human-Machine Systems

Sarah Miller, Alex Kirlik,
Alex Kosorukoff (University of Illinois)
and
Michael D. Byrne (Rice University)

Technical Report AHFD-04-17/NASA-04-6

December 2004

Prepared for

NASA Ames Research Center
Moffett Field, CA

Contract NASA NAG 2-1609

Abstract

Synthetic Vision System (SVS) displays are intended to improve aviation safety in low visibility conditions by providing the pilot with a computer-generated view of the terrain through any weather conditions. It is possible that an SVS system will sometimes be imperfect, providing the pilot with a display that does not match the environment outside the cockpit. This research is directed toward developing a model of visual attention allocation when the information contained in a display is imperfect. Prior models of visual attention allocation in human-machine systems assume that the information provided by an interface is always perfectly consistent with the information contained in the environment.

An experiment was designed and executed, based on Senders (1964) alarm detection paradigm and informed by the SEEV model of attention allocation (Wickens, Goh, Helleberg, Horrey, and Talleur, 2003), in which the informativeness of proximally available information presented on the display was varied in an effort to identify the potential effects of imperfect automation on visual attention allocation. Along with manipulating the informativeness of the display, the experiment manipulated bandwidth (BW), or how fast the information changes over time, and also value, which describes the importance of the information. Results presented in this thesis demonstrated that the informativeness (ecological validity) of the display had pronounced effects on visual attention allocation, and as such, this factor should be included in future models of visual attention allocation. Interestingly, the results also indicated that a slightly imperfect display presented a more difficult environment to the participant than a display providing even *less* informative information to the participant. This suggests that imperfect automation, especially if it is only slightly imperfect, may have detrimental consequences for monitoring in human-machine systems. Additionally bandwidth and value were found to best fit the data in an additive rather than a multiplicative formulation, contrasting with the current formulation of the SEEV model.

Table of Contents

List of Figures

List of Equations

List of Tables

Chapter 1

Introduction

Limited visibility is a common contributing factor in aviation accidents (Glaab and Takaluu, 2002). The introduction of Synthetic Vision System (SVS) displays into airplane cockpits is intended to improve safety in low visibility conditions by providing the pilot with a computer-generated view of the external scene through any weather conditions (Byrne and Kirlik, 2004). SVS displays are intended to reduce the frequency of visibility-related accidents by improving situation awareness, reducing controlled flight into terrain occurrences, and reduce low visibility loss of control events (Glaab and Takallu, 2002). The SVS system relies on GPS technology to know the exact location of the airplane and provides the pilot with a detailed map of the local terrain from an on-board database. Thus, the pilot is essentially supplied with an electronic picture of the outside world, as it would appear on a sunny day, though nothing may be visible outside the window but fog, rain, or darkness.

As with any new expensive technology, a considerable research effort has been focused on evaluating the impact of introducing SVS displays into the cockpit. The current research was performed to inform the design of a computational cognitive model of pilot cognition and behavior, which is intended to reduce the need to empirically evaluate each and every system design feature (see Byrne and Kirlik, 2004). Since visual attention has been found to be critical in monitoring flight deck automation (Parasuraman, Sheridan, and Wickens, 2000), one of the primary components involved in the current modeling effort was directed toward investigating visual attention allocation. Modeling attention allocation informed by eye-movement measures should assist in evaluating the effects of SVS displays because automation often has unintended negative side effects on scanning behavior (Parasuraman and Riley, 1997).

One of the problems that arose in the development of the computational cognitive model was determining how visual attention allocation was affected by monitoring imperfect automation (Parasuraman and Riley, 1997). It is possible that the SVS system will occasionally provide the pilot with a map of the terrain where the features are mismatched with the actual external scene. For example, the SVS may display an image where the runway is misaligned with the actual runway or may not show traffic that is present outside the aircraft (Byrne and Kirlik, 2004). The current research is directed toward informing a small piece of this modeling effort by focusing on developing a more comprehensive understanding of visual attention allocation when the SVS display provides imperfect information to the pilot.

Although there is an extensive body of research on visual attention allocation in the supervisory control domain (see e.g. Senders, 1964; Moray, 1986; Wickens, Goh, Helleberg, Horrey, and Talleur, 2003), a review of the literature, presented in Chapter 2, found that current models of visual attention allocation do not adequately address the reliability issue. All models assume that the information available from the interface is perfectly consistent with the information available from the environment. Due to this one-to-one mapping confound (Brunswik, 1956), it is unclear from the previous studies whether visual sampling was based on adaptation to the information provided by the interface, the environment, or if the object of adaptation is the relationship between the interface and the environment.

The current research is based on Senders (1964) alarm detection paradigm and informed by the SEEV model of attention allocation (Wickens, et al., 2003). Senders (1964) model postulated that visual attention allocation to a display was driven by the bandwidth (BW) of the information, or the rate of change of information contained on the display. The current study was informed by the SEEV model of visual attention allocation, which includes both a bandwidth and a value component. Value is related to the importance of the information contained on the display. According to the SEEV model, for experienced pilots, bandwidth and value should be the primary factors driving visual attention (Moray, 1986; Wickens, Helleberg, Goh, Xu, and Horrey, 2001). These previous models, however, were not able to address the issue of how attention allocation is affected when the interface does not provide reliable information about the environment because the BW was assumed to be a perfectly reliable indicator of the task criterion. Therefore, an additional factor that was manipulated in the current study was the informativeness of the BW cue in relation to the task-environmental criterion. The level of informativeness in this research was termed “ecological validity” following Brunswik’s (1956) definition of ecological validity as the correlation between a proximal cue and the distal referent.

A simple example describing how BW and value could be combined and mediated by the reliability or ecological validity of information is searching for a piece of information on the internet. If a person wants current news, she might visit the New York Times website (nytimes.com) rather than a sports website like espn.com because the former is updated every 10 minutes, while the latter is only updated every hour (i.e. BW is greater for nytimes.com) *and* she wants information about current world events rather than sports news (i.e. she places more value on current news rather than sports). Additionally, the Times website was chosen rather than a tabloid news site like eonline.com because the Times was found to provide a more informative or reliable source of news than the gossip website (i.e. it has a higher level of ecological validity).

In piloting, if the instrument displaying air speed changes faster over time than the instrument displaying heading, current models of visual attention allocation would predict that the air speed gauge should be sampled more often than the directional gyro relative to the difference in BW. However, BW does not necessarily correspond with the end goal of the monitoring task, which is often to detect when a gauge has gone out of range (i.e. alarm rate). It is possible that the airspeed gauge may have a high BW, but it may not travel into an unsafe region very often. On the other hand the directional gyro may not move very fast, but might travel into an unsafe region quite often.

It is unclear from the previous models how pilots will sample the areas of interest under these circumstances, because the models confound the proximal, or readily available, information contained in the interface (i.e. BW) with the distal (i.e. inferred through proximally available information) goal-directed task criterion (i.e. alarm rate). In order to investigate this issue, an experiment was designed and executed in which the informativeness of proximally available information presented on the display (i.e. the relationship between BW and alarm rate) was varied in an effort to identify the potential effects of imperfect automation on visual attention allocation. Chapter 3 discusses the methods of the current study in greater detail.

Chapters 4 and 5 present the experimental results. Chapter 4 describes the results associated with task performance data, while Chapter 5 details the modeling effort associated with the eye-movement data. In Chapter 5, we describe the application of Brunswik's Lens Model (1956) to visual attention allocation. To the best of our knowledge this is the first application of this modeling technique to visual attention (e.g. see Hammond and Stewart, 2000).

Chapter 6 summarizes and discusses the implications of these results in terms of the study's goals, which were to investigate the effects of imperfect displays on attention allocation and also to potentially inform the SEEV model. The results inform the experimental questions in two ways. First, BW and value were found to combine additively rather than a multiplicatively in mediating visual attention allocation, contrary to the original formulation of the SEEV model. Second, evidence presented in this thesis demonstrated that the informativeness, or ecological validity, of the display had pronounced effects on visual attention allocation, and as such, this factor should be included in future models of visual attention allocation. Interestingly, the results also indicated that a slightly imperfect display presented a more difficult environment to the participant than when display provided relatively poor information to the participant. This suggests that imperfect automation, especially if it is only slightly imperfect, may have detrimental consequences for monitoring in human-machine systems. Chapter 7 discusses these results in terms of implications for SVS display design.

Chapter 2

Background

Several factors are known to influence scanning behavior by guiding visual attention to various areas of interest in the aviation environment. These factors have been incorporated, often in the supervisory control literature, into several models of monitoring behavior. Monitoring behavior and supervisory control, although different, are often closely connected (Moray, 1986). Monitoring, or visual sampling, is a form of selective attention where the operator scans a display, without actively seeking to change the system state (Moray, 1986), in order to maximize value or minimize cost (Wickens and Hollands, 2000). These models are often found in the supervisory control literature. Monitoring behavior and supervisory control, although different, are often highly related because appropriate control requires a veridical understanding of the supervised system (Moray 1986). Monitoring is a necessary requirement for supervisory control, but it is only one of the components involved. Since visual scanning while piloting is a not a supervisory control task, but simply a monitoring task, this review will not cover the full range of supervisory control. Instead, this section will discuss the models that have been developed to predict attention allocation strategies in monitoring behavior in human-machine systems.

2.1 Fitts and Colleagues (initial research)

Some of the earliest studies to address monitoring behavior were conduced by Fitts and colleagues in the late 1940s and early 1950s (Fitts, Jones, and Milton, 1950; Moray, 1986). The authors recorded eye movements by over 40 pilots under various flight conditions. This series of studies were seminal in establishing the relationship between fixation frequency and duration of eye movements and the importance and difficulty of monitoring the instruments. Fitts summarized the findings:

> It is reasonable to assume that *frequency* of eye fixations is an indication of the relative *importance* of that instrument. The *length* of fixations, on the contrary, may more properly be considered as an indication of the relative *difficulty* of checking and interpreting particular instruments. A *pattern* of eye movements, i.e., the Link Values between the instruments, is a direct indication of the goodness of different panel *arrangements*. (Senders, 1983 p. 8)

Fitts further commented:

> Information about how pilots use their eyes while flying on instruments in fundamental to a basic understanding of the functions served by aircraft instruments and to simplification of the psychological processes that occur while a pilot is controlling an aircraft's altitude, location, and rates of movements in three-

> dimensional space. If we know where a pilot is looking, we do not necessarily know what he is thinking, but we know something of what he is thinking about. (Senders, 1983 p.9)

These results have been applied to redesigning the instrument layout in aviation cockpits (Senders, 1983). For example, as shown in Figure 1, the cross pointer was the most frequently fixated instrument. In the initial studies, this instrument was located far to the side on the panel. The instrument panel was redesigned so this instrument was located more centrally with respect to the other instruments (Fitts, et al., 1950).

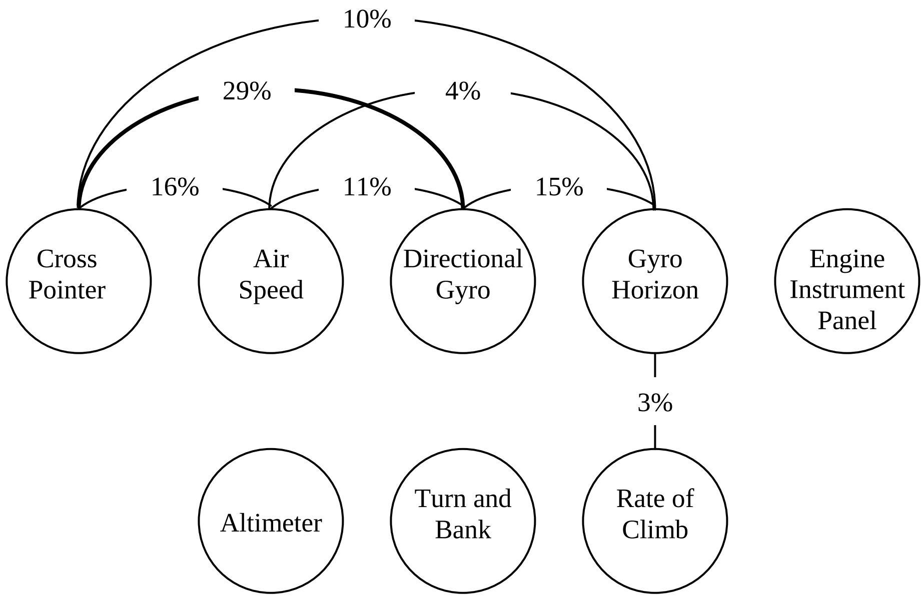


Figure 1. Eye movement link values from Fitts et al., (1950). Values less than 2% are omitted.

The studies by Fitts and colleagues (1950) were also informative in establishing an understanding of the general limitations of fixation frequencies and durations. Fitts et al., (1950) found that fixation duration during instrument monitoring while piloting was approximately 2 fixations per second.

Senders (1983) noted that while the Fitts and colleagues studies were valuable in improving instrument panel layouts, the results of the study were not generalizable to other aircraft, flight conditions, maneuvers, or instruments. The problem with the studies was that they only recorded pilot's eye movements without simultaneously recording the position or characteristics of the instruments. This one-sided relationship, looking only at the pilot, prevented Fitts and colleagues from making general predictions about instrument scanning. Although Fitts et al., (1950) assumed that new instrument arrangements would not change the behavior of the observers, Hopkin (1982) noted that introducing a new radar display to air traffic controllers completely changed their pattern of behavior in unexpected ways. Moray (1986) suggested that these results showed that the patterns could not have been predicted based on the quantitative data (as in

(Fitts, et al., 1950), but instead that the changes were qualitative. This suggests that a normative model of scanning could not be developed based on the Fitts and colleagues studies because of the lack of environmental data. Therefore, in order to draw valid conclusions from the data, it would be necessary to empirically evaluate each new instrument addition or display layout change.

To develop a model that was generalizable to other systems or conditions (Senders, 1983) suggested that a theory needed to be developed that considered the signal or environmental characteristics as well as the observer characteristics.

2.2 Senders' Model (bandwidth)

Senders (1964) provided one of the first quantitative models of operator sampling behavior (but see Craik, 1947 for an earlier approach). Senders approach was based on an information theoretic treatment of continuous functions that represent the signals that human-machine system operators must often monitor. Senders model assumes that operators behave as a classical Shannon communication channel. According to this theory, operators sample to reduce uncertainty about the state of the system in order to reconstruct the signal, which provides continuous information to the observer. The uncertainty in the system stems from the rate of information, or *bandwidth*, presented in the display. Senders (1964) applied the Nyquist sampling theorem, which suggests that, to effectively monitor a display, it is necessary to sample a signal at two times the bandwidth (a bandwidth of W Hz would need to be sampled at a rate of 2W Hz).

The goal of this work was to establish a quantitative relationship between observers behavior and the instruments they are monitoring (Senders, 1983). Senders (1983) stated the goal of the studies in simple terms: "Why does a pilot, confronted with many instruments on an instrument panel, look at instrument I at time T for duration D following a previous look at instrument I, X seconds ago? Any why does that pilot look at instrument I more often after looking at instrument J than after looking at instrument K." Further, Senders noted that pilots are often unaware of their own eye movements and believe that they are following the regular left to right successive sweeps across the instrument panel as they were taught.

The first study by Senders (and reexamined in Senders, 1983) was conducted using a simple paradigm. Senders (1964) tested the information-theoretic model by asking operators (n=4) to monitor a display with four instruments, one mounted at each corner of a square and separated by 60 degrees of visual angle. Each of the instruments had a different bandwidth ranging from 0.5 to 4 radians per second. The bandwidths were chosen so that operators could be able to continuously monitor the set of instruments, at a rate of 2W fixations per second, and would not be overloaded by the required observations. Operators were asked to monitor the four displays and indicate when any one of the instruments went out of range (a pointer moving into an "alarm" area).

It is important to note that the operators in this study, as in the majority of studies intended to inform the analysis and design of human-machine systems, were highly practiced. In the Senders (1964) study, operators were given over 30 hours of training to become adapted to the task. The pilots in the Fitts et al., studies most likely had hundreds if not thousands of hours of experience with their task.

As shown in Figure 2, Senders found that much of the variance in operators' visual scanning was explained by the bandwidth of the instrument. However, operators expressed a tendency to undersample high bandwidths and oversample low bandwidths. This is a consistent finding across many fields (see the discussion of "sluggish beta" in Wickens and Hollands, 2000). When reanalyzing the data, Senders (1983) was able to account for the undersampling by suggesting that, at high bandwidths, observers are able to detect bandwidth instantaneously. This means that it would only be necessary to sample at a rate of W instead of 2W Hz (Moray, 1986). Moray (1981) suggested instead that the oversampling was caused by forgetting.

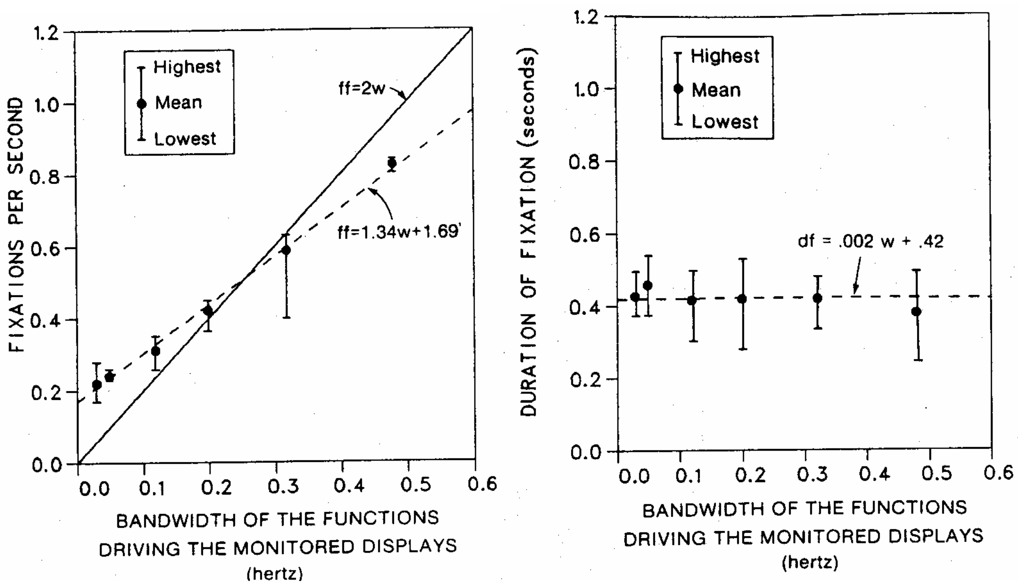


Figure 2. Results from Senders (1964) (adapted from Moray, 1986).

In addition to providing an understanding of sampling strategies as influenced by the bandwidth of the displayed information, the Senders (1964) data measured fixation durations. These results correspond with the Fitts studies, in that the dwell durations were fairly constant at around 0.6 seconds in the Fitts studies and 0.4 seconds in the Senders studies. Moray (1986) suggested that the Senders studies found shorter fixation durations because the studies took place in a laboratory and the Fitts studies were conducted in an airplane.

Senders was able to successfully apply his model of sampling behavior to the more applied driving domain (Senders, Kristofferson, Levison, Dietrich, and Ward, 1967), showing that the model may be generalizable to complex environments. That study proposed that in driving, scanning behavior is influenced by the curvature of the road, density of obstacles, the ability of the vehicle to maintain a straight line without correction, and the driver's estimation of the probability that other traffic will impede into the path of his or her vehicle. The authors briefly mention that due to the dynamic nature of the driving task (e.g. changing traffic, density of road signs, etc.) bandwidth is variable, leading to the need for drivers to update sampling strategies to accommodate changing bandwidth rates. However, the study did not suggest how changing bandwidth rates will affect driver's sampling strategies, or how much (if any) attention or other cognitive resources are required to update those sampling strategies.

(Moray, 1990) suggested that bandwidth is the primary factor guiding selective attention in transportation. He suggested that, in driving, the environment is highly structured and a glance at one state informs the driver about the state of the remaining objects in the neighborhood. This ecological structure substitutes for direct observation. Additionally, Moray (1990) suggested that bandwidth is limited while driving and although the vehicle is moving fast, signal bandwidths are slow relative to the limitations of the perceptual system. Since vehicle movements constrained by physics, the system is somewhat predictable merely through mass-induced vehicle dynamics.

Although Senders and colleagues (Senders 1964; Senders, et al., 1967) provided a starting point for understanding what drives visual attention allocation, there are some noted limitations of the model. The first criticism that has been levied against the Senders Model is that it is not representative of "real-life" tasks (Moray, 1986). In the Senders et al., experiments, the instruments were not given any meaning, therefore the operators (most likely) did not prescribe any value to any of the displays (but see Senders et al., 1967 for an application). It is indeed rare that all displays in human-machine systems are of equal importance. For example, the fuel gage has a high cost of a missed signal, but unless one drives past a police radar gun the speedometer often does not have a high cost for exceeding posted limits. Also, it is not realistic to assume that the operator will never be overloaded. Senders does not account for how operators will react to this overload (Carbonell, 1966).

Another limitation of the original Senders Model is that it does not account for situations where the instrument is near the "out-of-bounds" area when viewed (Moray, 1986). Intuition would suggest that if an operator has noticed a control is close to the limit, his or her next glance at that control should be made sooner than the model would suggest. An example of this in driving would be a driver noticing that her gas gauge is close to empty creating a situation where she might look at it more often. A complementary model was later proposed to account for this situation (Senders, 1983), but no studies were found that performed an empirical investigation to validate the models.

Finally, in a post-hoc evaluation of his initial studies, Senders (1983) raised some interesting observations about the applicability of theory to more complex tasks that could be re-interpreted to inform the current study:

> All of the experiments demonstrated that the sampling of low frequency signals was well above the theory when any of the theories was based upon stimulus generated uncertainty…If there is a minimum interval below which the monitor cannot go, then these data and the uncertainty threshold model which appear so rational for the restricted laboratory task cannot hold for the complex real situation. In the real world, a team of perhaps four observers may be confronted by an array of as many as two thousand indicators and signals. In order that they can cope with the task, the 'floor' created, apparently by forgetting, must be absent even with uniform intervals between observations and with uniform distribution of the monitoring load among the four monitors. There are 500 instruments per monitor which must be dealt with. If a

> monitor can look 2 ½ times per second there would be a delay of 200 seconds from one observation of any indicator to the next observation of that same indicator. (pp. 90-91)

Senders (1983) interpreted this result based on the assumption that observations were driven by maintaining an internal model of model of the task based on bandwidth. He suggested that:

> In a sufficiently reliable system, the exceedingly low failure probabilities provide only little experience with emergency requirements. The failure to observe many of the instruments frequently has not resulted in system failure or degradation. There is no cause for alarm in the loss of information from the low bandwidth signals that remain too long unobserved. Adaptation in the sense of an automatic and unconscious process may permit unawareness of overload by permitting a shift of performance criterion. This result would be, of course a dangerous one. The operators are lulled into a belief that all is well when in reality all is not well. It is merely that not enough time has passed to permit the full scope of events to occur to most operators. This leads to what might well be an aphorism of complex systems that *the more reliable the plant the less reliable the human operator*. (p. 91)

It could be suggested that Senders interpretation of the power plant situation may not be entirely correct. One could make the argument that operators were not adapting to the BW of the signal per se, but were in fact optimally adapting to the environment that the signals represented. More specifically, if the operators were sampling based on the task goal (e.g. to detect alarms), then optimal sampling should be based on knowledge about how often the gauge traveled into the alarm region and not on knowledge about how fast the gauge moved. So, it is possible that the operator did not appear to be overloaded because the operator wasn't overloaded. The reliability of the plant was part of the environment that was important for understanding sampling strategies. The environment that the plant operator is monitoring should not be discounted when developing a model of attention allocation. Senders (1983) could not differentiate between BW and the task criterion because he did not make the distinction between the proximal variables that are directly available to vision (via an interface) and the distal (environmental) variables, which are the ultimate object of adaptive behavior.

2.3 Carbonell's and Sheridan's Models (value)

Several studies have proposed modifications to Senders early model to make the model more representative by adding a value-based component (Carbonell, 1966; Sheridan, 1970; Kvalseth, 1977 (a)). Carbonell's Model (1966) adds a value component by adding a cost to missing a signal. Similarly, Kvalseth's Model (1977) explored adding a cost for errors (also added a sampling cost-based effort component). Sheridan (1970) added a positive reward-based value component for reducing error.

Carbonell (1966) suggested that different instruments involve various consequences of missing signals (e.g. missing fuel error more costly than speed). His model used a queuing theory-based approach, in which the central idea was that each instrument is competing for attention and the goal of the operator to minimize the cost of missed signals. Cost was calculated using Equation 1 and 2, based probabilities and costs of errors.

$$C(t) = \sum_{i=1}^{M} \frac{C_i P_i(t)}{1 - P_i(t)}$$

Equation 1. Carbonell's model of probabilities and cost.

The total cost of looking at an instrument at time t will be:

$$C_i^{'}(t) = C(t) - C_i P_i(t)$$

Equation 2. Carbonell's total cost equation.

Where:

M = number of instruments
t = observation time
C(t) = cost associated with exceeding threshold (time independent)
P(t) = probability that the instrument will exceed threshold at time t

Kvalseth (1977) developed a similar model, but added a component based on the cost of sampling (an effort component). Kvalseth examined monitoring behavior in extremely low-bandwidth processes (printed numerical displays), which is not similar to many of the faster dynamics encountered in aviation. The interesting addition in this formulation was the idea of combining sampling cost and cost of error into one model. However, a criticism of the experimental design is that it did not allow enough practice time for participants to obtain optimal sampling behavior (Moray, 1986).

Sheridan (1970) suggested that, instead of a cost associated with missing an event, there should be a reward for reducing error. Under this conception, the goal of the supervisor is to maximize return based on the value function and the bandwidth of the process. The optimal sampling interval, based on the net payoff, is calculated by taking the value of the last observation (positive value) minus the cost per unit time of not sampling (negative cost).

2.4 Bohnen, Leermakers, and Colleagues (decoupling BW and alarm rate)

Bohnen and colleagues recognized that the Senders studies (e.g. Senders, 1964) coupled BW with alarm rate and designed a series of studies to remedy this confound (Bohnen and Leermakers, 1991; Bohnen, Leermakers, and Venemans, 1996). The authors postulated that it was not necessary to reconstruct the signal in a monitoring task, as Senders suggested, because the goal of monitoring was to determine when a signal exceeded its limits. Thus, the authors proposed that visual attention allocation was based primarily on timing sampling to detect when a gauge would travel outside a permissible range. This timing was based on the local, proximally available, characteristics of the gauge such as the degree with which a sampled signal falls short of an alarm region combined with the immediate rate of change of the instrument. The authors suggested that global features, such as BW and alarm rate, were considered to be distal, inferred, characteristics that did not affect sampling when the local characteristics were most important (i.e. when the gauge was close to an alarm region).

Although Bohnen and colleagues (Bohnen and Leermakers, 1991; Bohnen, et al., 1996) studies were in the spirit of the Senders task (1964), their experimental design was quite different from Senders', so it will be briefly explained. First, the study decoupled BW from alarm rate by presenting the participant with the same number of alarms per gauge (4x8 alarms per 15 minute trial = .035 alarms/second), while varying the BWs of the gauges (0.03, 0.06, 0.09, and 0.09 Hz). Participants were presented with numeric displays where only one value could be queried at a time. The other three gauges were presented as blank windows. No eye movements were recorded, but sampling could be inferred from when the gauge was queried. Since participants could not infer the dynamics of the gauge based on a numeric display, rate of change information was presented with plusses or minuses (between one and three depending on the rate of change) below the value of the gauge. The goal of participants was to detect when a gauge traveled into the alarm region. A correct detection could occur at any time between when the gauge first traveled into the alarm region until 5 seconds after it left the alarm range. There was a reward for detecting alarms and costs for sampling and for missing alarms. The authors demonstrated that, when alarm frequency was removed as a confounding factor, BW was responsible for driving sampling. The local signal features (e.g. how close the gauge was to the alarm region) showed an influence on sampling.

In a follow on study Bohnen et al., (1996) further investigated the influence of alarm rate and BW on sampling rate. The experimental design was the same as Bohnen and Leermakers (1991), but the alarm rate differed across the gauges (0.03 Hz = 12 alarms, 0.03 Hz = 36 alarms, 0.09 Hz = 12 alarms, 0.09 Hz = 36 alarms). The primary goal of the study was to investigate the local signal characteristics on sampling behavior. The results suggest that sampling was influenced by the distance (as a function of BW and distance) the signal was from the alarm region. The results also demonstrated that sampling rate increased as a function of alarm rate, but the global features (e.g. BW, alarm rate) were dominated by the local features (e.g. time to out of bounds).

Although Bohnen and colleagues were insightful in their identification of, and attempt to rectify the confound between BW and alarm rate, there were several limitations of these studies that prevented further consideration with respect to the goals of the present research. First, the study suggested that sampling was not well described by Nyquist sampling theorem, but rather was driven by timing sampling based on the local characteristics of the gauges. However, the study design presented rate of change information numerically, which removed the need to repeatedly sample the gauge to infer the dynamics of the system. Also, the participant could detect an alarm from when the gauge traveled out of range until shortly after the gauge traveled back into the safe region, confounding the time available to detect an alarm with BW. Finally, there was a very high cost of sampling the gauges with a very low alarm rate (0.035 Hz in Bohnen and Leermakers, 1991). These studies did not find a strong connection between alarm rate and sampling, but it might be that the alarm rate was so low that it could not be readily inferred over the course of the study.

2.5 Other Mediators (effort, salience, habit, and contextual relevance)

Several other components have been found to contribute to attention allocation in scanning including effort, salience, habit, and contextual relevance (Wickens, et al., 2003).

The effort involved in accessing information has also been examined as a factor that potentially influences visual attention allocation. Generally, people try to conserve effort as much as possible, which suggests that highly effortful situations may influence optimal scanning behavior (Wickens, et al., 2003). Senders (1983) noted that, while frequency of fixations is due to the importance of the information, dwell durations may indicate the difficulty (or effort) involved with checking and interpreting the information. Kvalseth's (1977) model discussed in the previous section included an effort component as an attentional mediator. During a dual task situation, the effort component might come more into play because of already limited resources, but in highly trained pilots it is unlikely that effort will have a large influence on visual attention allocation (Moray, 1986).

Another factor that has been shown to influence scanning is salience, or the bottom-up attention-capturing ability of an event (Wickens, et al., 2001). Salient events are characterized by an increased ability to automatically capture attention. That said, salience does not always have predictable effects on visual attention allocation in human-machine systems. For example, (Mumaw, Sarter, and Wickens, 2001) found that pilots failed to notice the onset of an automation warning, while in basic attention research (e.g. Yantis, 1993), abrupt onsets have been found to readily capture attention. Also, Wickens et al., (2001) suggested that high salience may even reduce eye movements to an area of interest.

Habit and Context are two other factors that have been shown to influence scanning strategies (Wickens, Helleberg, and Xu, 2002). Influences on scanning behaviors due to habit might include learned behavior like pilot's scanning a display based on a "hub and spoke" pattern (Bellenkes, Wickens, and Kramer, 1997). In driving, habit might influence scanning to all road signs, even if the driver knows no relevant information will be contained in the sign. Contextual influences are based on the expected consequences of events that have just happened (Wickens, et al., 2001). For example, after seeing the driver in the vehicle ahead talking on a cell phone, one might expect that vehicle to drive more erratically.

2.6 The SEEV Model (salience, expectancy, effort, and value)

Recently, Wickens and colleagues combined several of the components involved in monitoring strategies and in attention allocation into the "SEEV" model of visual attention allocation (Wickens, et al., 2003). The SEEV model was designed to integrate the multiple mediators of attention allocation (discussed at length above) into a single model. This model suggests that attention is directed to an instrument or display based on four factors: Salience (S), Effort (EF), Expectancy (EX) (or bandwidth), and Value (V). Equation 3 reflects the functional form of the model, where P(A) is the probability of attending to a particular area of interest (AOI), and *s*, *ef*, *ex*, and *v*, are (estimated) parameters indicating the relative weighting or importance of the factors presumed to mediate attention allocation.

$$P(A) = sS - efEF + (exEX * vV)$$

Equation 3. SEEV Model.

Wickens et al., (2001) suggests that expectancy and value (expected value) are both top-down knowledge driven components, guided by an internal mental model of the environment. These two factors are considered to be the "optimal" factors in guiding attention allocation, that is, these two components *should be* the only factors that drive scanning. With well-trained operators, these factors should closely mirror actual observed scanning behavior (Moray, 1986).

They are not, however, the only components that influence attention allocation. Salience and effort are identified as two bottom-up factors that inhibit the optimal scanning pattern. Effort is described as an inhibitor to eye movements, while salience draws extra eye movements to a location. Wickens et al., (2001) suggests that in situations where operators are highly trained (as in Senders, 1964; Carbonell, 1966), these nuisance factors have a minimized effect on optimal scanning strategies.

This model was designed to integrate previous models of information seeking (e.g. Senders, 1964; Carbonell, 1966; Sheridan, 1970), which combined bandwidth and value on a one-to-one mapping, with models of task management applied to aviation (Raby and Wickens, 1994). Wickens et al., (2003) suggested that since pilots are well-trained with a well-calibrated mental model, the effort and salience components were not expected to have a large influence on scanning strategies. Therefore, a simplified model of visual attention allocation only included the expected value portion of the SEEV model was created.

In an effort to validate the simplified expected value portion of the SEEV model in a realistic environment, Wickens and colleagues conducted a series of studies (2001; 2002; 2003) with well-trained pilots. In order to develop a simplified model of visual attention allocation that could be applied to the aviation domain, Wickens and colleagues incorporated a hierarchical goal structure for assigning numerical values to the piloting tasks (i.e. aviate, navigate, communicate). Weights were assigned to the components in the model by first identifying the areas and tasks of interest and putting them into a matrix. Each cell in the matrix was assigned a priority (aviate higher priority than navigate which is a higher priority than communicate) based on the importance of directing attention to the location and event relative to the other cells in the matrix (Schutte and Trujillo, 1996). Numerical values were assigned to the cells using the lowest integer that kept ordinal values intact. This method was used in an attempt to simplify the model for easier application.

Experiments 1 and 2 compared scanning behavior in free-flight vs. baseline conditions (ATC input) as well as conflict vs. non-conflict trials. It was found that the optimal expected-value model accounted for approximately 80% of the variance in scanning. Experiment 4 added a communications task (task-relevant communications) to the baseline condition from experiments 1 and 2. The auditory input was modeled as a bandwidth parameter. The model accounted for approximately 95% of the variance in actual scanning behavior. Experiment 3 examined the effects of traffic density on scanning strategies. It was found that 95% of variance in scanning strategies was accounted by the EV model, suggesting that experienced pilots are able to adapt their scanning behavior well to changes in traffic levels and in a manner relatively uninfluenced by effort and salience.

A summary of the experiments conducted by Wickens et al., (2003) is presented in Figure 3. Figure 3 shows that the model predictions are quite consistent with the obtained dwell times, suggesting that well-trained pilots optimally adapt their scanning patterns to expectancy and value and are not as influenced by the nuisance factors (salience and effort). These findings are in agreement with Moray (1986), who suggested that well-trained pilots should have few departures from an optimal scanning pattern (e.g. BW and value).

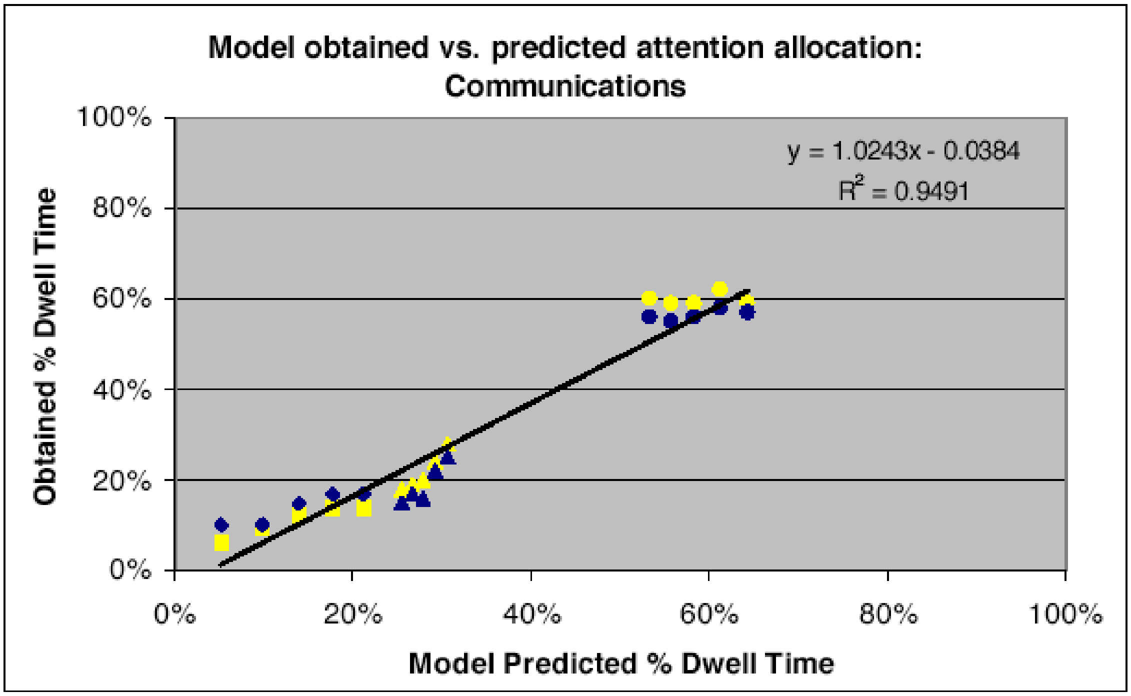


Figure 3. Model fit of predicted versus obtained dwell time based on expected value formulation of the SEEV model from Wickens et al., (2003). The various symbols represent different AOIs. The different conditions are represented by the light and dark shadings.

2.7 Summary and Limitations

The previous section of this thesis identified several factors that influence scanning behavior including bandwidth (or expectancy), value, effort, salience, habit, and context. The most influential factors, expectancy and value, have been modeled and validated in an aviation environment using the SEEV model (see Wickens, et al., 2003 for a summary). Although it is recognized that the SEEV model is not necessarily a model of optimal scanning, it should be a good predictor of visual attention allocation when the operator or pilot is well-trained.

However, up to this point the SEEV model, as described in Wickens et al., (2003), has been applied using only ordinal rather than ratio scale measurements of its parameters. In addition, the SEEV model does not appear to make provisions for situations when the proximal information available from various AOIs bears only probabilistic (fallible) relationship to the distal task criterion. Nevertheless, the idea that the two most dominant mediators of visual attention allocation are BW and value (as reflected in the SEEV model) provided a good starting point for framing our inquiry into the possible effects of imperfect information in human-machine systems.

Chapter 3

Methods

A gap was identified in the literature that needed to be addressed in order to understand and guide the visual attention allocation modeling of the NASA-SVS data (see Byrne and Kirlik, 2004). Several of the SVS conditions were concerned with understanding what happens to visual attention allocation when the SVS display does not give reliable information about the external environment. In order to more fully understand visual attention allocation under these circumstances, modeling required that the factors that guide attention allocation be defined. The previous models identified in the literature (see e.g. Senders, 1964, Wickens et al., 2003) have always predicted that the BW of the instrument is the primary driver of visual attention, but the underlying assumption behind these models is that BW is perfectly predictive of the end goal (e.g. detecting alarms). None of the models, with the possible exception of Bohnen and colleagues (1991 and 1996) have even suggested or made attempts to describe visual attention allocation when BW was not a perfectly reliable cue to the ultimate task-environmental criterion.

The primary goal of this study was to remedy this gap in the literature, resulting from the assumption that BW was always a perfectly reliable cue to predict the task criterion, by decoupling signal BW and the task criterion. In the current laboratory task, the goal, described in greater detail in a subsequent section, was to detect alarms. We design the task in such a way as to help clarify whether operators adapt to BW, alarm frequency, or to ecological validity (the correlation between the BW and alarm rate) (Byrne and Kirlik, 2004). A second objective of this study was to evaluate and potentially inform the SEEV model of visual attention allocation. This model suggests that, for experienced observers, visual attention is driven primarily by expectancy (BW) and value (Wickens et al., 2003). Therefore, the influence of value on attention allocation was also examined. In summary, *Goal 1* was to investigate the possible effects of EV on visual attention allocation, and *Goal 2* was to potentially inform the SEEV model by clarifying if the combination of the BW and value components should be additive or multiplicative.

Additionally, there were two sub-goal of this project. The first was to examine how observers learned to perform the task and how ecological validity (EV) affected learning. The second sub-goal was to examine if changes in EV, after learning had taken place, had any effect on visual attention allocation and performance. These sub-goals will be explained in greater detail in a following section.

3.1 Participants

Fifteen students from the University of Illinois at Urbana-Champaign participated in the study. Each participant completed five one-hour sessions over the course of five consecutive days and was paid $8 per hour for his or her participation. In addition to the $40, participants were given the opportunity to receive an additional $50 performance bonus for being the top performer in his or her experimental group (divided by EV condition as described below).

3.2 Display Design and Configuration

A sample experimental display is depicted below in Figure 4. The display contains four gauges, one at each of the corners of the monitor. The diameter of each gauge was 5.25 cm. The horizontal separation between the center of the gauges was 17.5 cm, and the vertical separation was 13 cm. Participants were positioned so that their eyes were 35 cm from the monitor. This resulted in a visual angle between the gauges of 28.1 degrees in the horizontal direction and 21 degrees in the vertical direction. This visual angle was chosen so that fixations were required to determine the exact position of the gauge.

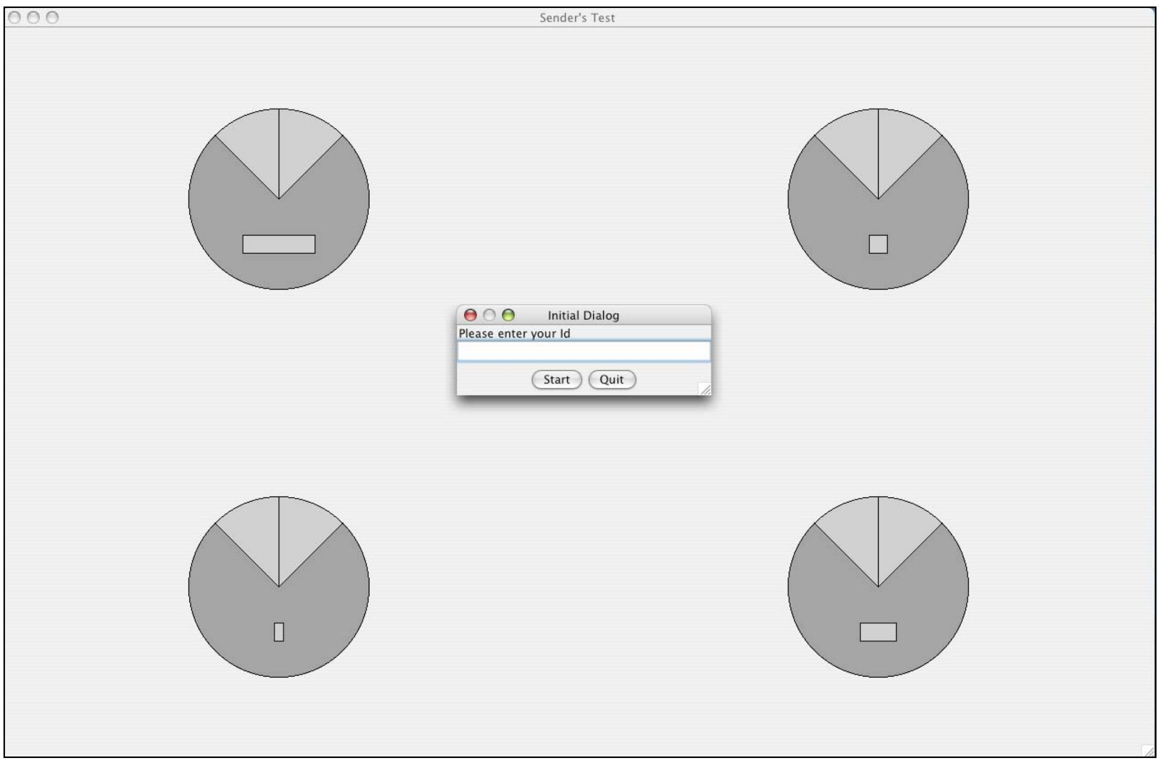


Figure 4. Sample display at trial onset. Note that this screen shot filled the screen on a Macintosh 17" flat panel studio display.

For each gauge, the dark gray area is the alarm or out of bounds region and the light gray was the safe range. The participant's goal was to detect when the pointer traveled into the alarm region (within one second of passing the threshold). The safe range was 90 degrees in circumference or "width", with 45 degrees on each side of vertical. At the onset of each trial, the pointer in each of the gauges was vertically centered in the safe range (shown as the vertical line at 12 o'clock above in Figure 4).

Directly below the safe region on each gauge the light gray bar represents the value prescribed to the gauge. The widest bar was worth 8 points, the bar half the size of the longest bar was given half the value of 4 points, the next smallest bar (again half the width of the 4-point bar) had a value of 2 points, and the smallest bar was worth 1 point. Therefore, in Figure 4 above, the upper left, upper right, lower left, and lower right gauges were valued at 8, 2, 1, and 4 points respectively.

In addition to assignment of a value, each gauge was assigned a bandwidth (BW), of 0.5, 1.0, 2.0, or 4.0 radians per second. Bandwidths were chosen to be identical to those used by Senders (1964). Recall from the previous section that Senders and colleagues chose these rates based on the Nyquist sampling theorem and Fitts' findings of the limitations of sampling rate (Senders, 1983). The Nyquist theorem suggests that to effectively monitor a display, operators would be required to sample a signal at two times the BW (a signal of W rad/sec would be sampled at 2W). It is assumed that, from Fitts data, operators could make approximately 2 fixations per second. Therefore, based on Senders assumption that monitoring is based on reducing uncertainty, the BWs were such that operators would be required to continuously monitor the display but not be overloaded by the requirements. Note that the values and bandwidths used in this study were chosen to have the same relation to one another on a ratio scale (e.g. each value is twice as much as the one preceding it), as is also the case with the signal BWs.

Alarm frequency, the number of times a pointer traveled into the alarm region per unit time, was also the same as in Senders (1964). Across the four gauges, alarm frequency was set to approximately 0.25 alarms per second. This means that, on average, a pointer on one of the four gauges traveled into the alarm region once every four seconds. Recall that in the Senders study, alarm frequency for each gauge was directly correlated with BW. So, the gauge with the highest BW (4.0 rad/sec) traveled out of bounds with the highest frequency and the gauge that was traveling at half the speed (2.0 rad/sec) traveled into the alarm region at half the frequency.

It is not clear, based on previous studies reviewed, whether observers were adapting to the task criterion (e.g. detecting alarms) itself or to the bandwidth of an AOI because the two were assumed to be in perfect correspondence. In order to address the confound perfectly correlated event frequency and bandwidth in previous models of attention allocation (e.g. Senders, 1964; Wickens, 2003), we introduced an additional variable termed ecological validity (EV). EV was defined by Brunswik (1956) as the degree of correlation between a proximal cue and the distal variable to which it is related. In this study, EV was calculated as the correlation between BW and alarm frequency. EV was not measured per gauge, but was calculated as the average correlation for all the gauges in a trial. The three EV values selected were 1.0, 0.75, and 0.25. Therefore, the EV=1.0 condition allows for direct comparison with Senders studies, while the EV=0.75 and EV=0.25 conditions will allow us to decouple the potentially separable effects of the salient proximal bandwidth cue and the distal task criterion of alarm frequency on visual attention allocation.

3.3 Generating the Signals

In order to create the EV conditions described previously, it was necessary to develop a set of signals to drive the gauges that fit the specified bandwidth (0.5, 1.0, 2.0, and 4.0 rad/sec), alarm frequency (0.25 alarms/sec), and ecological validity constraints (1.0, 0.75, and 0.25). In addition to the experimental constraints, three other constraints were established for the signal sets. First, the signals were made to appear random to the human observers, so that periodicity could not easily be used as an aid in detecting alarms. Next, all four gauges had to produce alarms to prevent participants from offloading the observation of a gauge without penalty. Finally, each time a pointer traveled into the alarm region, it traveled a minimum distance past the alarm boundary. This constraint was intended to allow the operator to distinguish between when the pointer passed the alarm threshold. The program developed to run this study appears in Appendix B.

The signal for each gauge was driven by the sum of three sine functions with a specific frequency, phase, and amplitude. In order to find a set of four such signals that satisfied all the experimental constraints, the frequency, phase, and amplitude parameters for each of the three sine functions comprising the signals were found through optimization. A genetic algorithm was developed specifically for this study to find a set of signals that satisfied all the constraints (Holland, 1992). This algorithm was designed to search a computational space in order to find a set of sine functions that minimized the Euclidean distance from a specified target point (as defined by the constraints). For the purposes of this search, BW was treated as a constraint (0.5, 1.0, 2.0, and 4.0), while the algorithm searched for solutions that fit the EV and alarm constraints.

More than 100 hours of CPU processing time were required to find the entire set of signals used in this study. Even with this degree of computation, it was not possible to find signal sets that perfectly satisfied the EV and alarm rate constraints as well as the other design constraints (e.g. all gauges produced alarms). Therefore, it was necessary to use signals that deviated slightly from the target. In addition, pilot testing revealed the possibility that participants might recognize patterns in the signals, so four different signal sets were generated for each EV condition. These four signal sets were presented to participants in a randomized order across trials (see Table 1).

Table 1. Actual alarm rates and ecological validities for each condition.

EV = 1 Condition	Alarm Rate (alarms/sec) per Signal Set			
	A1	B1	C1	D1
BW = 0.5	0.02	0.01	0.03	0.03
BW = 1.0	0.02	0.04	0.02	0.04
BW = 2.0	0.08	0.04	0.08	0.08
BW = 4.0	0.16	0.16	0.16	0.16
Actual Alarm Rate	**0.28**	**0.25**	**0.29**	**0.31**
Actual EV	**0.99**	**0.96**	**0.98**	**1.00**
Euclidean Distance	0.03	0.04	0.04	0.06

EV = 0.75 Condition	Alarm Rate (alarms/sec) per Signal Set			
	A75	B75	C75	D75
BW = 0.5	0.04	0.06	0.05	0.03
BW = 1.0	0.06	0.06	0.02	0.06
BW = 2.0	0.04	0.08	0.08	0.08
BW = 4.0	0.08	0.08	0.08	0.08
Actual Alarm Rate	**0.22**	**0.28**	**0.23**	**0.25**
Actual EV	**0.76**	**0.77**	**0.70**	**0.79**
Euclidean Distance	0.03	0.03	0.05	0.04

EV = 0.25 Condition	Alarm Rate (alarms/sec) per Signal Set			
	A25	B25	C25	D25
BW = 0.5	0.01	0.03	0.01	0.05
BW = 1.0	0.12	0.10	0.12	0.08
BW = 2.0	0.04	0.04	0.04	0.04
BW = 4.0	0.08	0.08	0.08	0.08
Actual Alarm Rate	**0.25**	**0.25**	**0.25**	**0.24**
Actual EV	**0.22**	**0.30**	**0.25**	**0.30**
Euclidean Distance	0.03	0.05	0.00	0.05

3.4 Procedure

Upon arrival at the laboratory, participants were briefed on the experimental procedure and signed an informed consent document. Participants were read task instructions (see Appendix A), which included instructions on how the task would be scored (as a function of the value and alarm rate of each gauge as explained in Section 3.6). Participants were also informed that the experiment would require an hour of their time each day for five consecutive days. Each day participants completed eight five-minute trials (40 trials over the course of the study) with a short break following each trial and a five-minute break after four trials. Participants were not informed about the purpose of the study except that they were to monitor the display and detect, with a keypress, when a gauge moved into the alarm area.

3.5 Experimental Design

The factors that were manipulated in this study included bandwidth, value, and ecological validity (the correlation between alarm frequency and BW). Bandwidth and value were varied across gauges while EV was a property of the entire set of four gauges within a trial and experimental group. The fifteen participants were randomly assigned to one of the three EV conditions (1.0, 0.75, or 0.25). Within each EV condition, each participant (N=5) was assigned a random value and BW combination per gauge (see Table 2). BW and value were assigned on a gauge-by-gauge basis, so each participant was presented with all four BWs (0.5, 1.0, 2.0, and 4.0) and all four values (1, 2, 4, and 8) in a given trial. The combinations of BWs and values for each of the gauges were randomized across participants. Once assigned, this combination of BWs and values was kept constant for a given participant throughout the study. For example, if gauge 1 was assigned BW = 2 and value = 8, she would see the same BW and value in gauge 1 for all the trials. Across EV groups, participants were yoked according to their BW/value combinations. Therefore, in each EV group, one participant saw the same BW/value combination as a participant in another EV group. Five participants were assigned to each EV condition, S01-S05 were assigned to the EV=1, S06-S10 were in EV=0.75, and S11-S15 were assigned to EV=0.25.

Table 2. Experimental design.

	Gauge 0 (upper left)		**Gauge 1 (upper right)**		**Gauge 2 (lower left)**		**Gauge 3 (lower right)**	
	BW	Value	BW	Value	BW	Value	BW	Value
S01 S06 S11	0.5	8	1	2	2	1	4	4
S02 S07 S12	1	4	2	8	4	2	0.5	1
S03 S08 S13	2	2	0.5	4	1	8	4	1
S04 S09 S14	4	8	1	1	0.5	2	2	4
S05 S09 S14	0.5	1	4	8	2	4	1	2

Recall from the previous section that the signals generated in this trial were not completely random (as sum of the three sine functions). As such, four different signal sets were generated for each EV condition (see Table 1 in Section 3.2). The presentation of each signal set was randomized per trial across days (see Table 3 below), so all participants in an EV group saw the same signal set in a given trial and each participant observed the same signal set two times per day (10 times total).

Table 3. Signal set randomization.

	T1	T2	T3	T4	T5	T6	T7	T8
Day 1	A	B	C	D	C	D	B	A
Day 2	D	A	C	B	A	C	D	B
Day 3	C	A	B	D	B	C	A	D
Day 4	B	D	A	C	D	A	B	C
Day 5	D	B	C	A	A	B	C	D

Finally, we wanted to evaluate the effects of presenting participants with abrupt changes of the ecological validity of the task. The final four trials on day 5 (T36-T40) consisted of a transfer manipulation, where the ecological validity of the task environment was modified. Participants who were assigned to the EV=1 condition in the first 36 trials were exposed to the EV=0.75 condition for the final four trials, the EV=0.75 group was transferred to EV=0.25, and those who originally performed in the EV=0.25 condition saw an environment with EV=1 for the final four trials. Participants were not told about this manipulation until the end of the study. Unless otherwise noted, all results and discussion will refer to the pre-transfer trials (T1-T36).

3.6 Measures Collected

The two types of dependent measures collected were performance and eye movement measures. Performance measures were recorded as a function of the participant's manual responses (key presses) generated per trial. Additional data was collected that recorded the position of the gauges and the time that elapsed since the start of the trial. These measures were recorded at a rate of 20 Hz. This data file was used to assess hits, misses, and false alarms. Eye movements were recorded using an Apple iSight camera. Fixation frequencies and durations were manually coded using a computer program specifically developed for this study. These data were coded to allow the eye movements to be matched with the position of the gauges at the time of each fixation.

Performance was translated into a total score for each trial, and calculated as the sum of hits on a gauge times the value for that gauge minus misses and false alarms multiplied by the value for each gauge (see Equation 4). In Equation 4, G0 –G3 indicate the summation over the four gauges.

$$TotalScore = \sum_{G0}^{G3} (Hits - (Misses + FAs))xValue_{Gi}$$

Equation 4. Total score equation.

A hit was defined as correctly detecting (by pressing the key that corresponded to the gauge) when a pointer moved into an alarm region, within one second of the pointer passing the alarm threshold. A miss occurred when an alarm traveled into the alarm region and it was not detected within the one-second window. A false alarm (FA) was recorded each time one of three conditions was met. First, a FA was recorded if a key for a corresponding alarm was pressed when the pointer was not in the alarm region or if a key was pressed when more than one second had passed since the pointer moved into the alarm region. There was only one opportunity to receive points each time a gauge traveled into the alarm region, so pressing the key that corresponded to the alarming gauge more than once would result in one hit. The remaining key presses were recorded as FAs. Since there was essentially an unlimited potential to lose points (misses and false alarm), but only a finite potential to receive points, it was possible (even likely in the beginning) to receive a negative score for a given trial.

It was necessary to normalize the score for each trial before comparing performance across trials or participants because the total score possible for each trial was different both between participants within trials, and within participants between trials. The total achievable score varied because of the various alarm rate and value combinations to which participants were exposed. For example, one participant's combination might include a high alarm rate coupled with a high value where another might be in an environment where a gauge with a high value is coupled with a low alarm rate. The total achievable score differed within subjects between trials because the different signal streams generated for each EV condition had slightly different alarm rates for a given gauge (see Table 3.

Scores were normalized by taking the score received for a trial divided by the total score possible for that trial (see Equation 5).

$$TotalScore(\%) = \sum_{G0}^{G3} \frac{(Hits - (Misses + FAs))xValue_{Gi}}{TotalHitsPossiblexValue_{Gi}}$$

Equation 5. Normalized total score equation.

The total score possible, in Equation 5, was calculated as the number of alarms at a given gauge multiplied by the value for that gauge and summed across all four gauges (G0 through G3). Thus, the maximum score per trial was 1.0 (100%), which means that all alarms were successfully detected and there were no false alarms (misses are the compliment of hits). There was no minimum score possible because false alarms subtracted points from the score, so it was theoretically possible to achieve a high negative score. Note that hits could also be analyzed using a similar method, but false alarms could not be normalized according to this formula because there was no maximum number of false alarms possible per trial.

Chapter 4

Performance Results

The results are presented in two chapters: a chapter describing the performance results (Chapter 4) and a modeling chapter (Chapter 5). The current chapter details the results from the performance measures, which are essentially a record of the manual responses (or key presses) of the participant in reference to the state of the signal. The performance measures were presented as a function of value and the percentage of hits, misses, and false alarms as described in Chapter 3. All the results presented in this chapter should be assumed to be pre-transfer trials (T1-T36) unless otherwise specified as post-transfer trials. Post transfer trials were T37-T40.

Data from 16 out of a possible 600 trials were lost due to occasional computer program crashes, which were not systematic. Lost trials were not included in the performance results.

4.1 General Overview

A general overview of performance over the course of the study is presented in Figure 5. Figure 5 illustrates the effect of the EV manipulation on performance in terms of total score achieved (see Equation 5 in the previous chapter). Each point on the graph represents the average total score for the five participants in each EV condition over all eight trials (except on day 5 where the two points represent the average of the 4 pre-transfer and the 4 post-transfer trials respectively) completed each day. Table 4 presents the numerical averages represented in Figure 5.

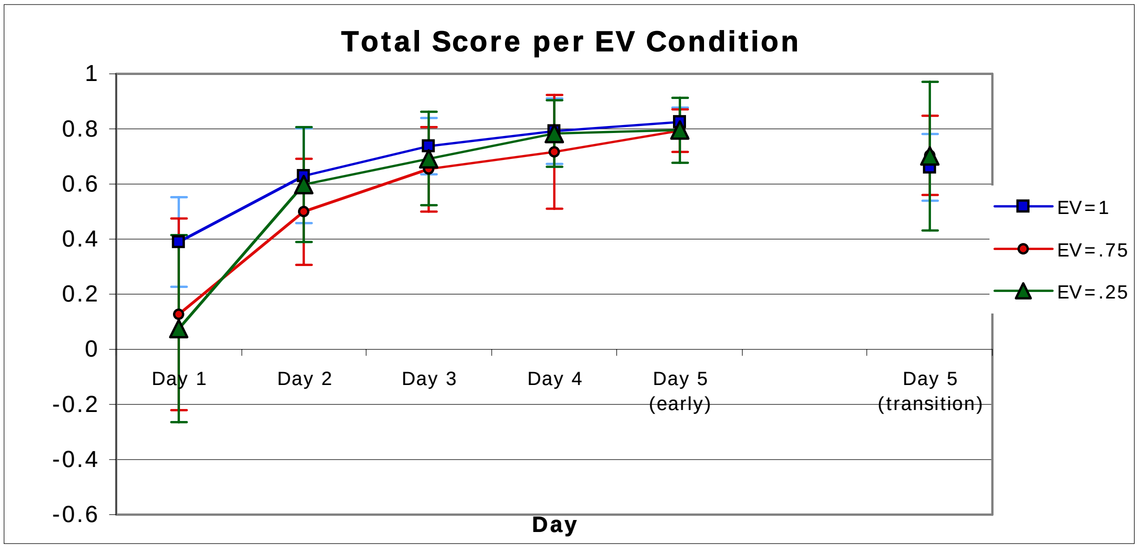


Figure 5. Total score by EV condition (mean per day and participant). The total score is a function of the total points received divided by the total points possible. The error bars indicate one standard deviation. Day 5 (early) indicates the pre-transfer condition and Day 5 (transition) indicates the post-transition results.

Table 4. Total score by EV condition (averaged across day and participant).

	Day 1	Day 2	Day 3	Day 4	Day 5 (pre-transition)	Day 5 (post-transition)
EV=1	0.39	0.63	0.74	0.79	0.82	0.66
EV=.75	0.13	0.50	0.65	0.72	0.79	0.70
EV=.25	0.07	0.60	0.69	0.78	0.80	0.70

Figure 5 illustrates several potentially interesting points that warrant further analysis. First, performance in general, as measured by the percent of points received divided by the total achievable score, increased significantly over the course of the study across all EV conditions ($F_{(33,396)} = 24.330$, p<0.01). The learning curve, shown by an increase in total score across the days, was steepest in the beginning of the study, and leveled out at around 80% by the final day for all three EV conditions.

Although all three EV conditions exhibited a learning curve, Figure 5 illustrates an apparent difference in the curves between the EV<1 conditions and the EV=1 condition. Analysis revealed that EV had a strong influence on learning, as indicated by a significant EV x trial interaction ($F_{(66,396)} = 1.521$, p<0.05). The EV=1 manipulation, where participants were exposed to an environment in which alarm rate and BW were perfectly correlated, performance appeared to be better in the early trials than in the other EV conditions. The performance difference is especially striking in day 1, where the total score percent for EV=1 is more than twice as high as the two EV<1 conditions (EV=0.75 and EV=0.25). In terms of learning, the significant interaction effect suggests that the participants in the EV=1 condition had less to learn over the remainder of the study than the two EV<1 conditions.

However, a main effect comparison among EV conditions did not prove to be significant ($F_{(2,12)} = 1.237$, p=0.325). It is possible that this finding is due somewhat to low power (n=5 per condition). Further evaluation of the potential effects of EV will be provided in the modeling analysis in Chapter 5.

Additionally, recall that different BW and value pairs were presented to participants in a given EV condition in order to decouple any potential effects on scanning behavior associated with particular settings of BW and value. Essentially, we wanted to ensure that the BW and value combinations were randomly assigned. Reassuringly, particular combinations of BW and value did not account for a significant portion of the variance in scanning strategies.

4.1.1 A Comparison of Hits Versus Misses

Since the total score was calculated as a combination of points received from hits and points lost from misses and false alarms, it was possible that hits and false alarms could contribute differently to the total score. Therefore, the contribution of both hits and false alarms were examined independently per EV condition to determine how each of the factors may have individually influenced overall performance. Since misses are simply the compliment of hits (each time there was an alarm it could be detected or not detected), data on misses are not presented.

Figure 6 presents the hit percentage data, while Figure 7 shows the *number* of false alarms per EV condition. For Figures 5 through 7, the data were averaged for all five participants in each condition and for all the trials each day.

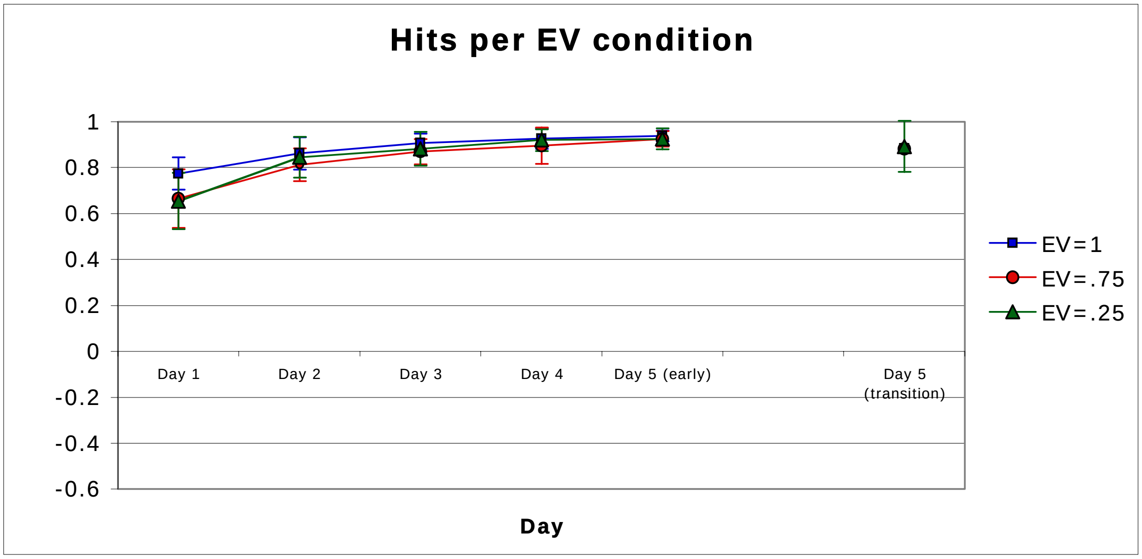


Figure 6. Hits (%) by EV condition (mean per day and participant). The error bars indicate one standard deviation. Day 5 (early) indicates the pre-transfer condition and Day 5 (transition) indicates the post-transition results.

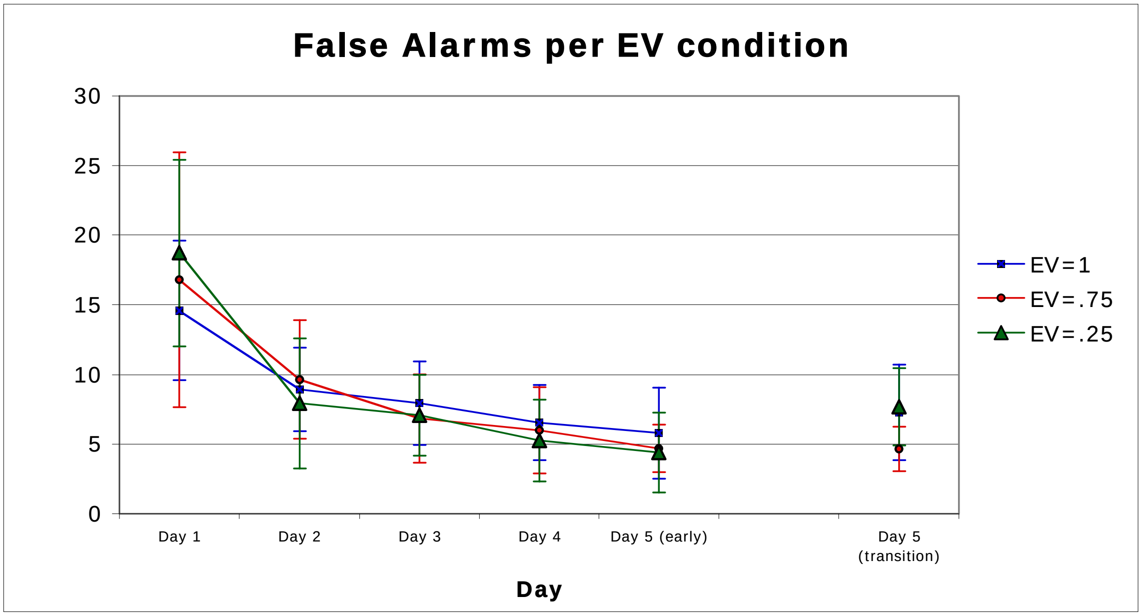


Figure 7. False alarms (count) by EVcondition (mean per day and participant). The error bars indicate one standard deviation. Day 5 (early) indicates the pre-transfer condition and Day 5 (transition) indicates the post-transition results.

Figure 5 and Figure 6, comparing total score and hits respectively, reveal a similar learning curve across days and appear to be proportionally similar with respect to the performance by EV condition. Notice that, in the EV=1 condition, performance was better throughout the study for total score and hits. In addition, Figure 7 shows a reduction in false alarms over time for all EV conditions over time, but false alarms were not significantly different across EV conditions. These results are in line with expectations, because it was assumed that any effects due to the EV manipulation would be revealed through hits rather than through false alarms. False alarms were not expected to be influenced by ecological validity because they are essentially errors of commission. Since false alarms did not significantly contribute to differences in performance across EV conditions, the remaining discussion will focus only on performance in terms of total score. In sum; the EV manipulation had its primary effect in moderating the rate at which participants could acquire the skill of generating hits as opposed to misses, but did not influence the rate at which false alarms decreased over trials.

4.1.2 Total Score Per EV Condition

Figures 8 through 10 break down performance in each EV condition by participant in order to shed light on potential individual differences within a given EV condition. In each of the three figures, the solid line indicates the average performance of the participants in the condition.

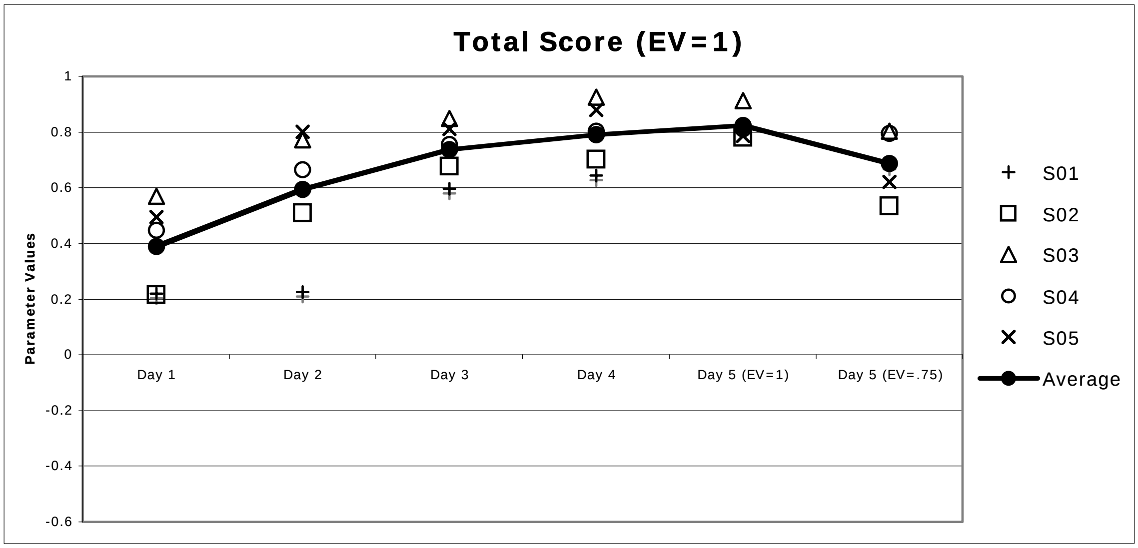


Figure 8. Total score per participant (EV=1) averaged across day.

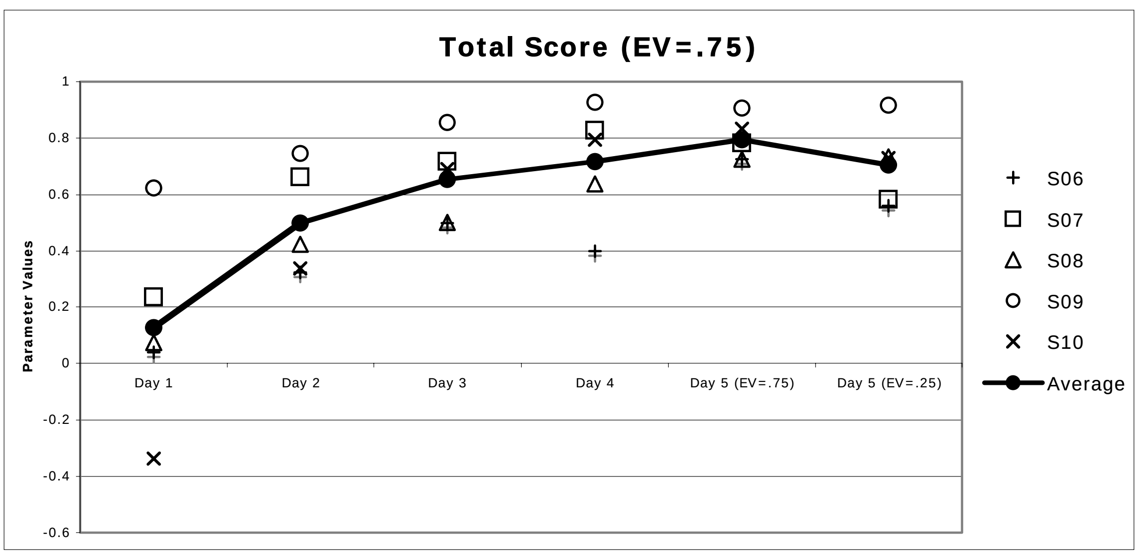


Figure 9. Total score per participant (EV=0.75) averaged across day.

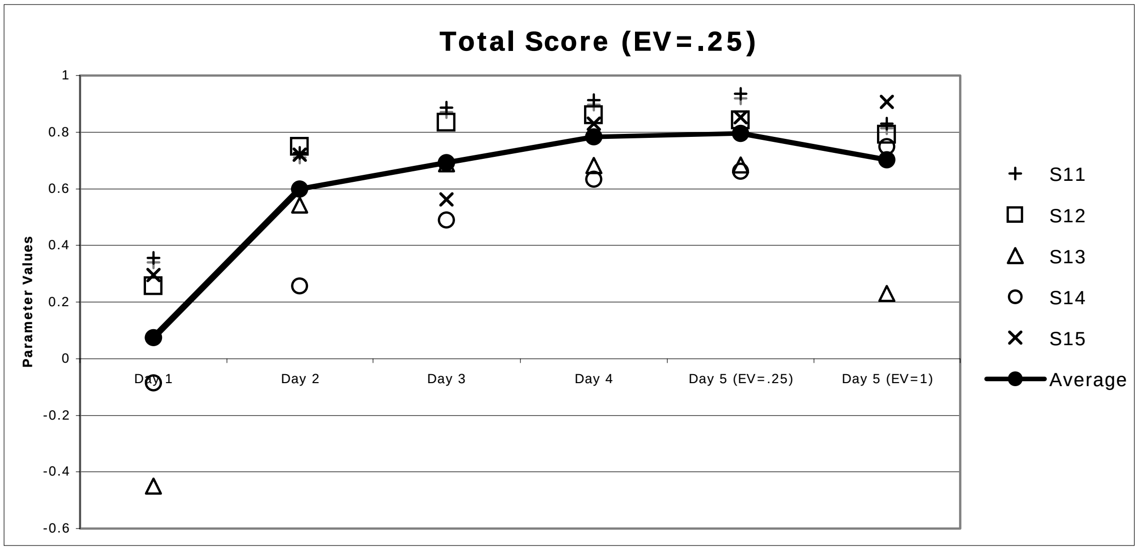


Figure 10. Total score per participant (EV=0.25) averaged across day.

Individual differences can be examined in terms of variance across participants within a given EV condition or between EV conditions. Within each EV condition, shown in Figures 8 through 10, it appears that individuals exhibited more variability in their ability to perform the task rather than in learning the task. In general, participants who started out as better (or worse) than average performers initially appeared to perform better (or worse) than average over the course of the study. Between EV conditions, it appears that individual differences were more varied in the two EV<1 conditions than in the EV=1 condition. The difference in ranges is especially pronounced early in the study. For example, in the EV=0.75 and EV=0.25 conditions, the range of total scores for day 1 (T1-T8) was from -34 to 62% (range=96%) for EV=0.75 and from -45 to36% (range=81%) for EV=0.25, while the total scores in the EV=1 condition ranged from 22 to 57% (range=35%). By the final four trials on day 5 (T33-T36), the variance in total scores was reduced for all the EV conditions from 39 to 92% (range=53%) for EV=0.75, from 66 to 94% (range=28%) for EV=0.25, and from 64 to 92% (range=28%) for EV=1. Notice that, although the variance was reduced across all EV conditions, the reduction was not equivalent. In the EV=0.75 condition, there was almost twice as much variance across participants than in the EV=1 and EV=0.25 conditions over the final 4 trials on day 5.

4.2 Early Trials

Since most learning appeared to take place early in the study, as evidenced by the steepest slope occurring between day 1 and day 2 in Figure 5, the data were broken down by trial for the early trials (T1-T10). Figure 11 presents total score performance for each EV condition per trial for early trials (T1-T10). Remember that eight five-minute trials were conducted per day, so Figure 11 presents data from all of day 1 (T1-8) and the beginning of day 2 (T9-10). This figure illustrates a steep slope, or learning curve, especially in the very early trials. Figure 11 also shows differences among EV conditions, especially when comparing the EV=1 and the two EV<1 conditions. Participants in the EV=1 condition outperformed the two EV<1 conditions from the first through tenth trial. However, Figure 11 shows that there does not appear to be much difference between the EV=0.75 and EV=0.25 conditions.

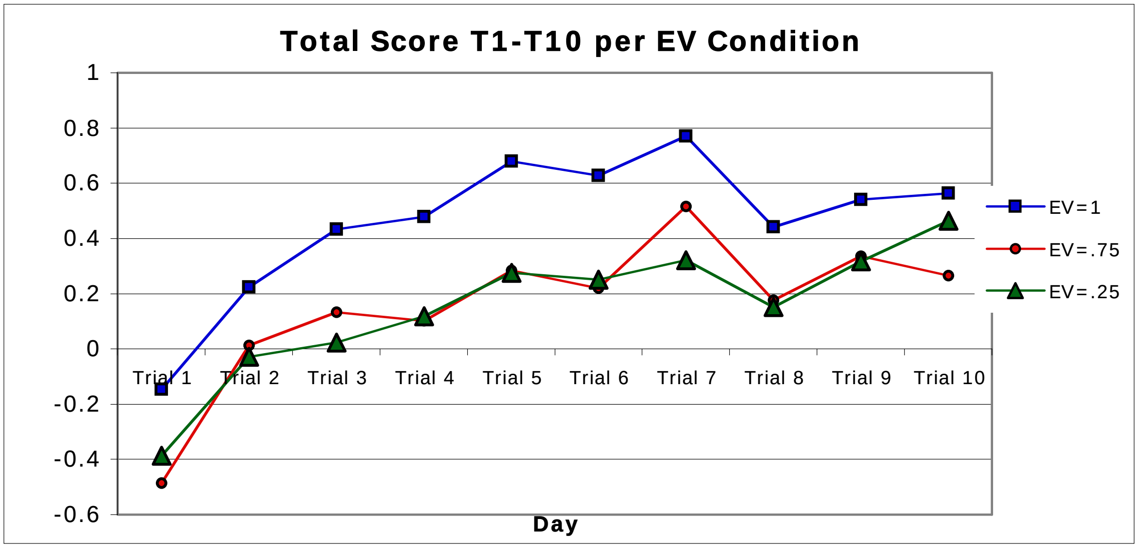


Figure 11. Total score for early trials by ecological validity condition (mean participant scores per EV condition)

4.3 Late Pre-Transfer Trials (T34-T36)

The final pre-transfer analysis examines the relationship among the EV conditions over the final set of trials (T34-T36). Although performance, in terms of total score, was relatively similar across EV conditions by the final trials, this does not necessarily suggest that participants were similarly adapted to their environment across the EV conditions. Figure 12 provides an alternative look at adaptation through examining misses according to the particular BW and value combination on a gauge. One could make the argument that better adaptation would be illustrated in a higher miss rate for gauges that have low value and BW combination rather than a high value for the BW and value combination because misses on the low value gauges represent fewer points lost than misses on the high-value gauges. Therefore, for the best adapted participants, one would expect to see a negative slope on a graph comparing miss rate with a BW/value combination. According to this logic, one would also expect steeper negative slopes to be indicative of better adaptation than flatter slopes. In the SEEV model, value and BW were combined as a product, but evidence from the eye movement data explained in the following chapter (Chapter 5, Section 2) suggested that these values should be combined additively rather than multiplicatively.

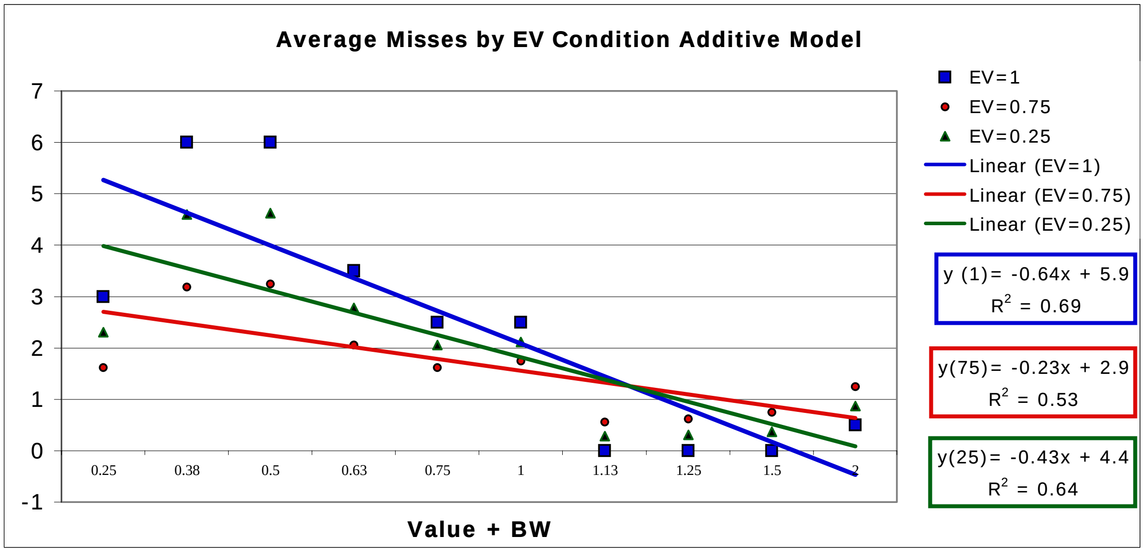


Figure 12. Average misses by EV (additive combination of BW and value) for T34-T36.

Figure 12 shows that the steepest slope is found for the EV=1 condition, followed by the EV=0.25 condition and EV=0.75 condition. Though all EV conditions show a negative slope, the EV=0.25 had a 33% steeper slope than the EV=0.75 condition, while the EV=1 condition had around a 33% steeper slope than the EV=0.25 condition. This provides evidence to suggest that participants in the EV=1 condition were more adapted to the task than the EV=0.25 and EV=0.75 groups. This graph also suggests, if the assumptions are correct, that the EV=0.25 group was better adapted to the environment than the EV=0.75 manipulation. This is an interesting difference across EV conditions, because at this stage in the study (T34-T36) there was very little difference in total score across EV conditions.

4.4 Post-Transfer Trials (T37-T40)

For the last four trials on the final day participants were exposed to a transfer condition, where the ecological validity was manipulated. This allowed us to investigate how changing the reliability of the proximal BW cue with respect to the distal task criterion (alarm frequency) affected performance. The participants who adapted to the EV=1 condition were exposed to EV=0.75, the EV=0.75 group performed in EV=0.25, and the EV=0.25 condition transferred to EV=1. Figure 13 graphs the final two trials prior to the transfer (T35-T36) and the final two post-transfer trials (T39-T40). Note that the graph does not include the first two trials immediately following the transfer condition.

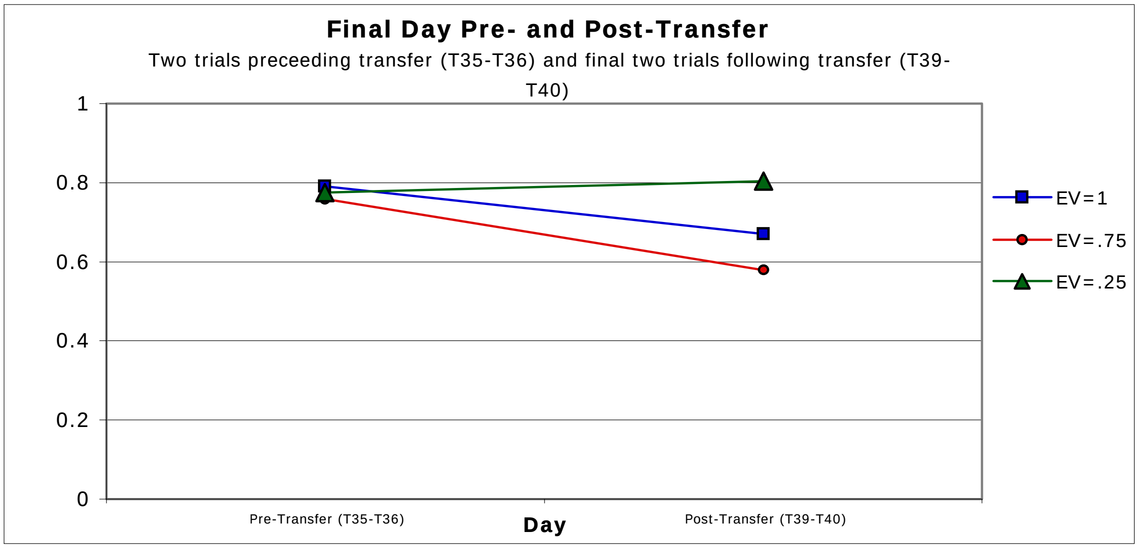


Figure 13. Final day final two trials pre-transfer (T35-T36) and final two trials post-transfer (T39-T40).

From Figure 13, it appears that participants in the EV=0.25 condition were able to maintain performance when the environment was manipulated, while participants in the EV=1 and EV= 0.75 conditions were unable to maintain performance at pre-transfer levels. Although Figure 13 appears to show an effect of EV, there was little statistical evidence to support this finding with a main effect of p=0.16 and an even higher p-value for the interaction effect. Since there were only five participants in each group, the apparent lack of significance could be explained in part by a small N and an outlier in the EV=0.25 group. Table 5 shows that S13 is more than two standard deviations from the mean. However, since there was no significant difference among EV conditions after the transfer, we will not provide further analysis of the transfer condition.

Table 5. Total score post-transfer for the final two trials in EV=0.25.

Participant	Total Score T39	Total Score T40
S11	0.79	0.98
S12	0.84	0.87
S13	**0.28**	**0.55**
S14	0.91	0.88
S15	0.97	0.98
Average	0.76	0.85

Chapter 5

Modeling and Results

Although performance measures were informative in telling part of the story of how bandwidth and value are mediated by ecological validity to guide visual attention allocation, performance is not a direct indicator of visual attention. Therefore, to paint a more complete picture of visual attention allocation, as it relates to the current study, eye movement data were analyzed and modeled. Since one of the primary goals of the thesis was to investigate how the informativeness of the proximal environment affected visual attention allocation, a complete model of visual attention allocation requires an understanding of how participants are adapting *to* the ecology. Brunswik's Lens Model (Brunswik, 1956) allowed us to develop a more comprehensive understanding of visual attention allocation as it relates to probabilistic functionalism, or the relationship between the *participant* and the informativeness of the *environment* (i.e. ecological validity) (Hammond and Stewart, 2001).

To the best of our knowledge this is the first application of the Lens Modeling technique (Brunswik, 1956) to the study of visual attention allocation. Therefore, the first section of this chapter provides a detailed review of the concept of probabilistic functionalism as it is represented in the Lens Model and as it applies to visual attention allocation. The final three sections in this chapter (Sections 2, 3, and 4) describe the results of the modeling analyses. Sections 2 and 3 focus only on the participant, or policy capturing, models. This means that these two sections do not incorporate the environmental considerations into the modeling analysis. Section 2 describes an effort by Byrne and Kirlik (2004) that investigated the most optimal method of combining the BW and value components in the SEEV model. Section 3 describes an unsuccessful effort to model the data to predict individual eye movements. Finally, Section 4 develops both the environmental and participant models and combines them using the Lens Model.

5.1 Lens Model Background

Though Brunswik's Lens Model has been predominantly applied as a tool in judgment analysis (Cooksey, 1996), the design of the model makes it a particularly useful tool in understanding how the components of this visual attention allocation task interact. Though this particular extension of the Lens Model has not been used in the attention domain, it shows promise for helping to determine how EV mediates the effects of the bandwidth and value cues on visual attention allocation. Since the Lens Model has not previously been applied to this domain, this section will describe the theory behind the Lens Model, underline the basic features of the model as they relate to the theory, and discuss how those features connect with the current extension into the visual attention allocation domain.

The foundation of the Lens Model rests in an approach Brunswik coined probabilistic functionalism (Brunswik, 1956). The basic concept of probabilistic functionalism is twofold: (1) it emphasizes the relationship between the participant and the environment; and (2) the participant-environment relationship is based on uncertain relationships among the environmental variables (Cooksey, 1996). The Lens Model is a tool for combining the various concepts of probabilistic functionalism into a single symmetrical representation. Therefore, one can see how the imperfect relationship between BW and alarm rate, as investigated in this thesis, are related to the concept of probabilistic functionalism and represented using the Lens Model.

Cooksey (1996) describes how probabilistic functionalism can be represented in various features of the Lens Model (see Figure 14). The first concept of probabilistic functionalism emphasizes the need to consider the environment or task in which the person is involved *along with* the traditional emphasis on sampling subjects. The key to recognizing the impact of the environment on behavior is founded in an understanding that people are adapting their behavior to a distal ecology in which they are situated by way of proximal (salient, readily available) cues ($X_1 ... X_n$). The relationship between the proximal cues and the distal criterion (Y_e) and the relationship between the person and the proximal cues (Y_s) is illustrated using dual symmetrical models. Brunswik (1956) suggested that the relationship between the proximal cues and the environmental criterion was not perfectly reliable, but only probabilistically related ($r_{e,1} ... r_{e,n}$). This relationship was termed ecological validity. Note that this is the second time ecological validity has been defined in this thesis. Ecological validity was also a dependent variable describing the correlation between BW and alarm rate. Here, it describes the relationship between all the proximal cues and the environment. Mirrored on the other side of the model is an imperfect relationship between the cues and the person, termed cue utilization. Brunswik suggested that the relationship between the person and the cues were also not perfectly reliable, but rather probabilistically related ($r_{S,1} ... r_{S,n}$).

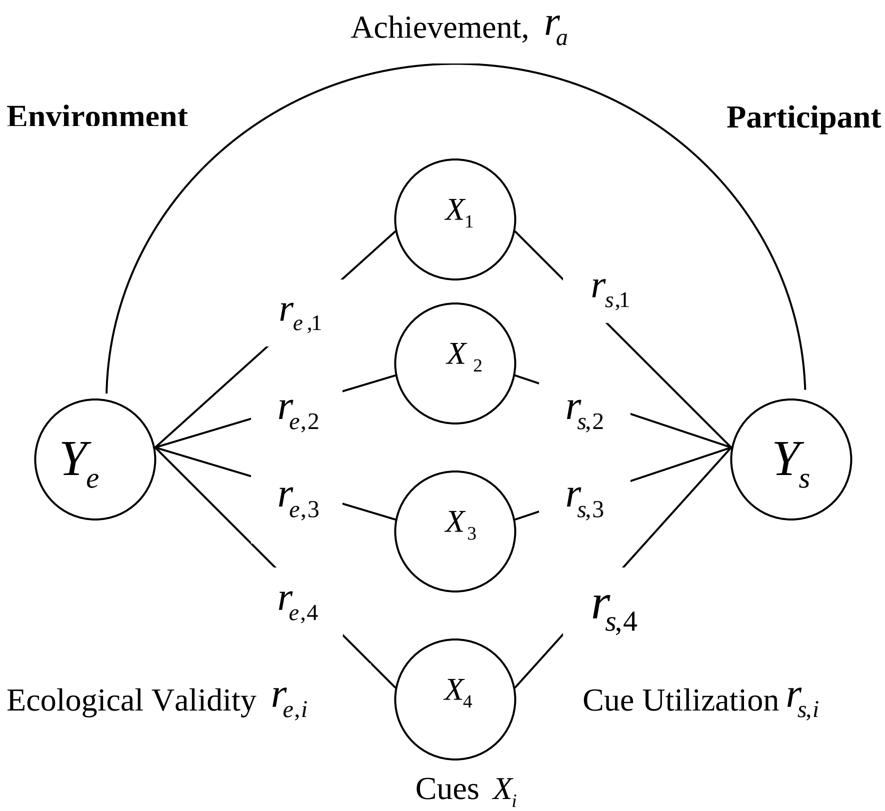


Figure 14. Brunswik's Lens Model (adapted from Bisantz, Kirlik, Gay, Phipps, Walker, and Fisk, 2000).

The two models (environment/cues and cues/person) are connected through the proximal cues, which are the same for both models. Additionally, the person and environment are related through an overall measure of achievement (r_a). Achievement is the degree to which the person adapted to the task criterion (environment) in relation to the degree it was possible to adapt to the environment. In other words, successful adaptation in the Lens Model is not measured in relation to performing perfectly with respect to the task criterion, but rather it is measured as the ability to adapt to the distal criterion based on the proximal cues available to the participant. This means that if the task criterion can not be predicted based on the cues available to the person, as is the case with imperfect automation, even a flawless performer’s achievement would be limited by the inherent uncertainty in the task.

This is an important consideration in the current task, because the relationship between the proximal cue of BW and the distal task criterion (i.e. alarm rate) was less than perfect in two of the three conditions. Thus, the Lens Model allows us to investigate both how the environmental cues *should* be used as well as how these cues *were* used across conditions.

Assuming the linear regression is used to create both human and environmental models by regressing behavior and the criterion on the cues (respectively), the connection between all the variables presented in the Lens Model can be formally compared in a quantitative form (see Equation 6 below).

$$r_a = GR_sR_e + C\sqrt{1-R_s^2}\sqrt{1-R_e^2}$$

Equation 6. Lens Model Equation (LME).

Where:

r_a	=	achievement
G	=	knowledge or the correlation between the participant and the environment
R_e	=	predictability of the environment based on the proximal cues
R_s	=	participant's consistency
C	=	unmodeled knowledge

The following example will attempt to further clarify the relationship between the Lens Model parameters by describing how the model was used in the present study. Figure 15 illustrates the Lens Model used in the current study. Although this figure has a few more variables, it follows the same general framework as Figure 14. The left side of the figure represents the environmental model and the right side represents the participant's model. The proximal cues used for the modeling were bandwidth and value.

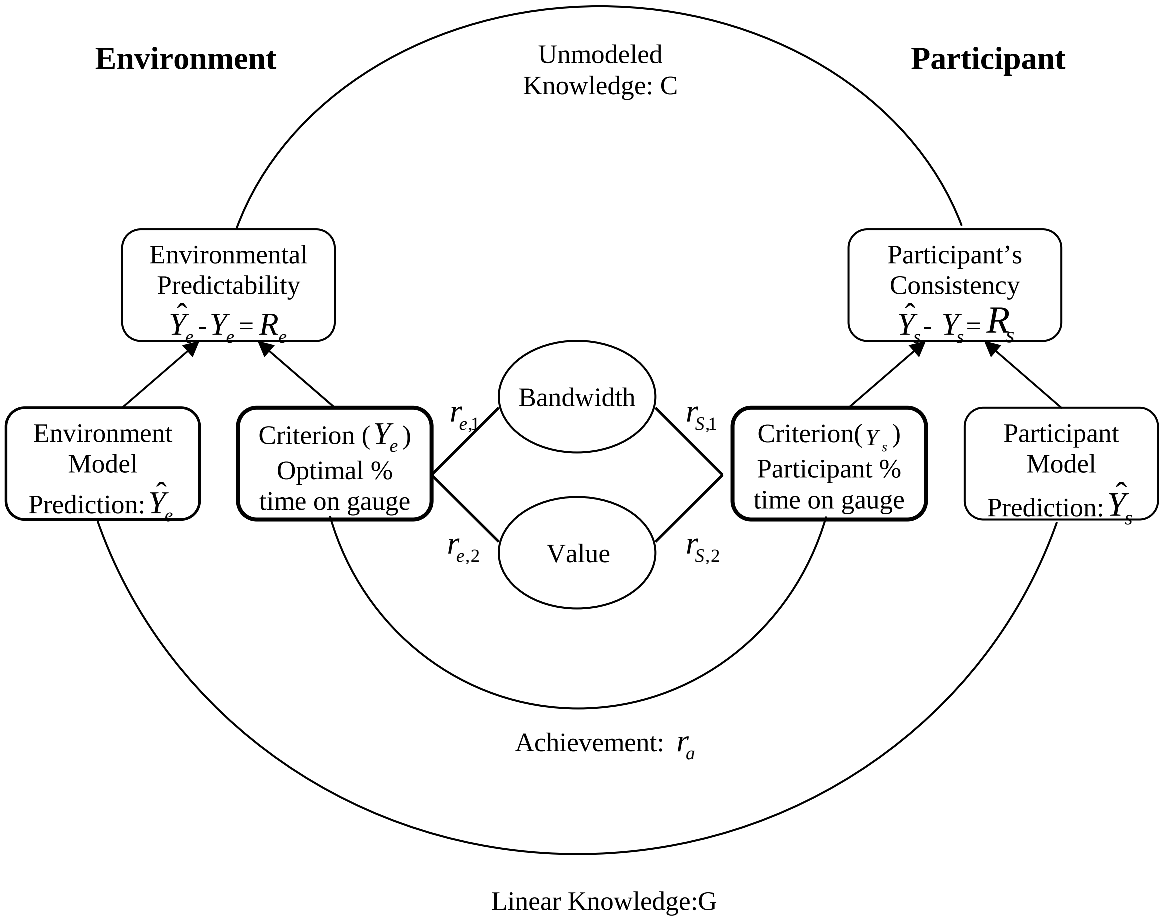


Figure 15. Lens Model depiction of visual attention allocation. (adapted from Bisantz, et al., 2000).

Table 6. Variables in the Lens Model Equation.

Variable	Name	Definition
r_a	Achievement	Correlation between the optimal and actual percent time on gauge.
R_e	Environmental Predictability	Correlation between the optimal percent time on gauge and the available cues. Measures how well the optimal percent time on gauge can be predicted based on a linear combination of the bandwidth and value cues.
R_s	Participant's Consistency	Correlation between the observed percent time on gauge and the available cues. Measures the consistency with which participant’s implemented a visual scanning strategy based on a linear combination of the bandwidth and value cues.
G	Linear Knowledge	Correlation between the predictions of the environmental and participant’s models. Measures the degree to which the optimal and participant’s beta weights for the bandwidth and value cues match.
C	Unmodeled Knowledge	Correlations between the residuals of the environmental and participant models. Measures the non-linear knowledge the participant had about the task that wasn’t captured by the linear model.
$r_{e,1}$, $r_{e,2}$	Optimal Beta Weights	The optimal beta weights for the bandwidth and value cues.
$r_{S,1}$, $r_{S,2}$	Participant’s Beta Weights	Participant’s beta weights for the bandwidth and value cues.

The environmental model (Y_e in Figure 15) estimates the environmental predictability (R_e), which is described as the relationship between the optimal criterion and the cues. R_e measures how well the optimal criterion can be predicted based on a linear combination of the available cues (BW, value). The optimal criterion, for this task, was the optimal *percent of time* that should be spent looking at each gauge, calculated based on the total score possible for each gauge. A high value of R_e, indicates that the BW and value cues can be linearly combined to be highly predictive of the optimal percent time on each gauge. There were several assumptions made in order to calculate R_e described in greater detail in a following section (Section 5.4.1).

The right side of Figure 15 depicts the participant’s model. This model measures the relationship between the actual observed percent time on each gauge (R_s) and the proximal BW and value cues. R_s, parallel to R_e, measures the consistency with which each participant implemented his or her scanning strategy based on a linear combination of the available cues (BW and value). Thus, a higher value of R_s represents a more consistent scanning strategy.

Three additional parameters (r_a, G, and C) link components in the participant’s and environmental models. Y_e and Y_s are compared through r_a, or achievement, which measures the correlation between the participant’s scanning strategy and the optimal scanning strategy. A high value of r_a means that the actual percent time spent on each gauge was similar in proportion to the optimal (environmental prediction) percent time on gauge.

A second link between the participant and environmental models is G, or knowledge. G links the predictions of the environmental model with the predictions of the participant’s model. Since the predictions for each model are based on the linear combination of the proximally available cues (BW and value), G represents the degree to which the beta weights of the two models match proportionally. G does not measure the absolute match between the beta values, but only the degree to which they are proportional. Thus, a higher value of G would represent a situation where the optimal beta weights for the BW and value cues were similar, in proportion, to the participant's beta weights for the cues. Cooksey (1996) defines G as knowledge of the task, where a value of 1 represents perfect knowledge of the task and a value of 0 represents no knowledge of the task.

Finally, C captures the unmodeled relationship between the participant and the environment. C can be interpreted in as (1) the extent to which the participant's non-linear weighting of the cues matches the non-linear structure of the cues in the environment, or (2) the extent to which the cues used by the participant are not represented in the model. A high value of C indicates that the participant has some (correct) knowledge about the environment that contributed to performance, but wasn't captured by linear modeling.

An alternate formulation (Stewart and Lusk 1994) of this model, called the Extended Lens Model or ELM, has been developed to compensate for the inability of the traditional Lens Model to account for errors of magnitude and scale in the achievement component.

This extended model decomposes skill into three components; a baseline achievement correlation (r_a), a regression bias, and a base rate bias. This more general formulation, termed Skill Score, shown in Equation 7 below (see Stewart and Lusk, 1994; Strauss and Kirlik, in press), allows achievement to be further analyzed to determine if these other factors can explain any apparent discrepancy between human behavior and an environmental criterion.

$$\text{Skill Score} = \text{achievement } (r_a^{\,2}) - \text{Scale Error} - \text{Magnitude Error}$$

Equation 7. Skill Score decomposition.

A base rate bias would be illustrated by the regression line crossing the x- or y-axis above or below 0. The regression bias would be observed as the degree to which the individual correlation points fall above or below the regression line providing the best fit between behavior and the criterion (i.e. the degree of scattering of the points). In the traditional y=mx+b equation, m is the regression bias and b is the base rate bias. Regression and base rate bias are somewhat connected in this study, because the time on each gauge has to add up to 1.0 across the four gauges. This means that, for each individual trial, the greater the base rate bias, the flatter the search slope has to be. These points are only somewhat connected because this connection only holds true for each individual trial, whereas the figures illustrate multiple trials for all the participants in a condition.

5.2 Investigation of SEEV as Additive vs. Multiplicative Model

One of the goals of this thesis was to potentially inform the SEEV model of visual attention allocation in a less complex environment than previous studies (e.g. Wickens et al., 2003). Recall that the most recent formulation of the SEEV model combines the expectancy and value components multiplicatively. In an earlier report using the data from the current experiment, Byrne and Kirlik (2004) investigated whether an additive model provided a better fit to the eye movement data than the multiplicative model. This regression modeling was intended to capture only the participant’s side of the model, so we do not discuss any potential influence of ecological validity on visual attention allocation in this section.

Byrne and Kirlik (2004) analyzed the coded eye movement data from an early trial (Trial 2) and a late trial (Trial 34) by calculating the percent of time fixated on each gauge as a function of bandwidth (expectancy in the SEEV model) and value. In order to examine BW and value as predictors of percent time on gauge, the authors first normalized the variables on a 0 to 1 scale. A regression model was created for each EV group (1.0, 0.75, and 0.25) for both the early and late trials. The regression model attempted to fit percent fixation per gauge as a function of bandwidth and value as either an additive (BW + value) or a multiplicative (BW x value) function. Thus, there were twelve regression models created in total: 6 models for the early trial (one additive and one multiplicative for each EV condition) and 6 for the late trial (again, one additive and one multiplicative for each EV condition).

As part of the experimental design (see Table 2), the BW and value were counterbalanced between participants in each condition. As part of this design, the value and BW were unintentionally perfectly correlated for one person in each group (see S05, S10, S15). This participant could not be modeled in the regression analysis because of this colinearity.

In contrast to Wickens et al., (2003), Byrne and Kirlik (2004) found the additive (BW + value) model provided a better fit (higher R^2 and adjusted R^2) for the additive rather than multiplicative model for each of the 6 comparisons (2 trial x 3 EV). Also, 5 of the 6 additive models explained a significant ($p<0.05$) portion of the variance in dwell percentage, while 3 of the 6 multiplicative models were significant. This means that the additive model provided a better fit for the eye movement data both early and late in the study.

Since Byrne and Kirlik (2004) found that the additive model provided a better fit of the data, the remaining discussion will focus on the additive formulation. Figure 16 illustrates the contrast between the R^2 values for the early (T2) and the late trial (T34) among the EV conditions. The late trial in the EV=0.25 condition was the only model that did not explain a significant portion of the variance.

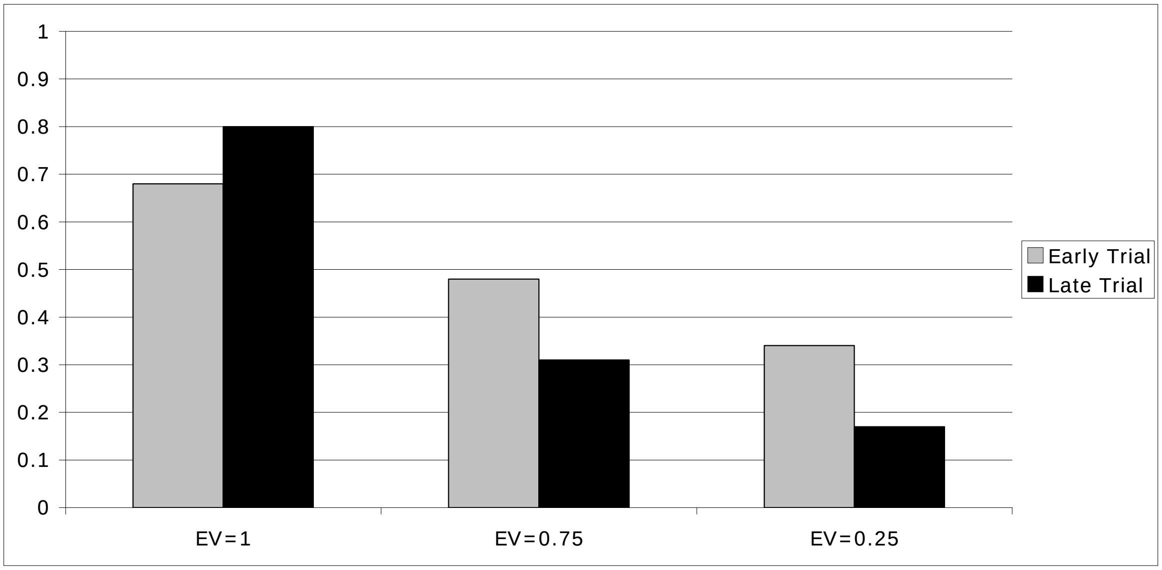


Figure 16. R^2 **values for linear regression predicting dwell time from bandwidth and value.**

5.3 Detailed Regression Modeling

The next area of interest in the analysis was to investigate whether we could develop models to predict where participants were directing visual attention, at the individual fixation level, based on any additional cues beyond BW and value. For this analysis, it was necessary to examine eye movements at a finer grain of detail than the per-trial level discussed in the previous section. This section will detail the analysis of attention allocation per individual fixations. The fixations were matched with signal properties at each transition point to try and determine if any other factors can be used to predict attention allocation.

In order to determine if any other factors were guiding attention allocation, the problem was framed using a decision metaphor. The goal was to try and predict the gauge to which a transition occurred based on the cues available to the participant. For each fixation it was assumed that there were three possible gauges a participant could transition to, because no transitions returned to the gauge that was currently fixated on. Thus, from a single gauge the transition probability would be (1, 0, 0), (0, 1, 0), or (0, 0, 1), where 1 is the probability that that particular gauge was transitioned to and 0 represents the gauges that were not fixated on. These transition probabilities represent the dependent variable in the regression analysis.

The independent variables in the analysis were designed to be the potential cues the participant was using to decide what gauge to make the transition to. These variables we identified as possible fixation predictors included:

(a) Value
(b) Bandwidth
(c) Alarm rate
(d) Distance the gauge was from the alarm region (normalized)
(e) Time until the gauge would enter the nearest alarm region based on the average BW

(f) Time until the gauge would enter the alarm region based on the current BW
(g) Time until the gauge would enter the alarm region based on the BW over the previous 0.25 seconds
(h) Time until the gauge would enter the alarm region based on the BW over the previous 0.50 seconds
(i) Time until the gauge would enter the alarm region based on BW and direction
(j) Time elapsed since last fixation
(k) Time elapsed since last fixation minus time to alarm at last fixation

In summary, for each transition (or observation) there were three rows of data (one for each gauge that the eye could transition to). One row was coded with a value of 1 in the dependent variable. This was the gauge to which that the participant actually transitioned. The two remaining rows were coded with values of zero in the dependent variable. These were the gauges that participants did not choose. The independent variables were coded separately for each row (gauge) based on the descriptions provided above.

Each participant was modeled separately and all participants in each EV condition were pooled into a single model for each trial. Although many factors were found to predict fixations, the best regression fit (R^2) for any participant model was under R^2=0.08. Since there was little evidence that visual attention allocation could be predicted per fixation, this analysis was not investigated further.

5.4 Lens Modeling and Regression Analysis

Lens Modeling was employed as a tool to address one of the primary and one of the sub-goals of the thesis. The primary goal addressed in this modeling analysis was to investigate how ecological validity (or the reliability of the information contained in the display) in addition to BW and value affected visual attention allocation. One of the sub-goals addressed in this section was to investigate how EV affected learning the task.

The performance results, detailed in Chapter 4, suggested that the majority of learning occurred very early in the study, as evidenced by the improvement in total score. The steepest slope of the performance curve occurred between the first and second day, so the first ten trials of the study were coded and analyzed. Recall that participants completed eight five-minute trials per day, so the early modeling analysis was taken from day one and the first two trials of day 2.

In order to address the final influence of EV on visual attention allocation, the second modeling effort was directed toward monitoring eye movements late in the study. For this analysis, the final three trials prior to the transfer condition were analyzed (T34-T36). Though it is possible, and quite likely according to Senders (1983), that participants are not as optimally adapted to their environment as they could be after many more hours of practice, Senders noted that sampling frequency should be stable by this point in the study. Modeling the late trials should uncover any existing differences between EV conditions that weren’t captured by the performance data. Recall that performance peaked out at around 80% for all conditions, but some initial performance results suggested that there were still some differences in adaptation between the conditions.

Looking at trials very early (beginning of day 1), late early (between day 1 and day 2), and late (final day) in the study should allow a comparison to be made both about how learning occurs over time and how it is affected by EV. Additionally, late trials (T34-T36) should allow us to examine final environmental adaptation both within and between conditions.

Additionally, we wanted to determine if Lens Modeling could be successfully extended to informing the visual attention domain. Though this was not one of the original goals of the study, this modeling technique presented itself as an untested tool for analyzing performance outside the judgment and decision making domain. If this tool proves to be useful for modeling this type of task, it presents an alternative for understanding visual attention allocation as well as provides another domain where the Lens Model can be successfully applied.

5.4.1 Assumptions and Considerations in Constructing the Models

In order to develop the Lens Model for this task, it was necessary to make several assumptions.

First, to calculate the optimum time that should be spent on each gauge for the environmental model, it was assumed that the percent time spent on each gauge should be proportional to the number of points that could be received by detecting alarms (i.e. total score). Remember that points were calculated by multiplying the value by the alarm rate. The optimal percent time on each gauge was calculated as a *multiplicative* function, but the Lens Model fits these cues as a linear (*additive*) model. This difference between the multiplicative and additive formulations could explain why performance peaked at about 85% by the end of the study. That is, performance may have peaked at 85% because the best linear additive cue combination strategy could have been only 85% successful at mimicking the true multiplicative way in which cues were combined in the environment. This modeling technique allows this hypothesis to be analyzed. Recall that the additive combination of the BW and value cues did indeed provide the best fit to actual eye movements (see Section 5.2 for a discussion). Thus, in the environmental side of the equation, R_e is not indeed truly optimal, but the best fit of the optimal percent dwell time on each gauge based on a linear combination of BW and value.

Another assumption required for the environmental side of the model was related to the level of detail in the environmental structure participants were adapting to over the course of the study. Since participants were exposed to four different signal sets with slightly different EVs and alarm rates, they were essentially presented with one of four possible environments each trial. Since the signal sets were randomized, it was unlikely that participants were predicting and adapting to the exact structure of a given signal set each trial. Rather, it was more likely that learning (as indicated by higher performance scores over the course of the study) was based on the global, stable, properties of the environment. One of the goals of the study was to examine the general influences of BW, value, and EV on attention allocation. So, it was assumed that each trial did not present the exact environment participants were asked to adapt to with respect to EV and alarm rate, but rather each individual trial represented one instance or sample of these factors. For example, in one condition, EV was taken to be 1, whereas EV on a given trial actually varied between 0.96 and 1.

Only one environmental model was developed per participant, rather than developing an independent model for each trial. This model was calculated by taking the optimal percent time on each gauge for each of the four signal sets (multiplied by the given value per gauge) and running a linear regression for all combined signals based on BW and value for each gauge. An average optimal model was also generated for each EV condition, which included all four participants that could be used per condition. This general model was calculated as the single best linear fit of all signal sets (4 per participant x 4 participants).

For the participants' models, it was necessary to combine three trials per model in order to have enough observations to explain a significant portion of the variance with the model. We did recognize, especially early in the study, that combining over three consecutive trials might wash out some of the changes in scanning strategies due to learning, but combining across trials within participants was assumed to be more appropriate in this case than combining across participants within trials because of the potential inter-subject differences were greater than the intra-subject differences.

For this analysis, only the only participants included in the discussion that follows were those were a significant portion of the variance could be explained by the regression analysis ($p<.05$). It was expected that fewer participants will have fewer reliable models early in the study because eye movements were most likely not as stable early rather than late in the study.

5.4.2 Environmental Models

The study was designed to sample three different levels of ecological validity (EV=1, 0.75, and 0.25). Therefore, it would be expected that the environmental models for the three conditions would differ with respect to both the R_e, or environmental predictability and the beta weights for the bandwidth cue. Since the value cue was not designed to contribute differently across EV conditions, this value was expected to be the same across the models for the EV conditions. Figures 17-19 illustrate the beta values for each participant's environment in each EV condition. Since each participant in a given EV condition was exposed to a different BW and Value combination, as explained earlier, it was necessary to run a regression analysis for each individual environment.

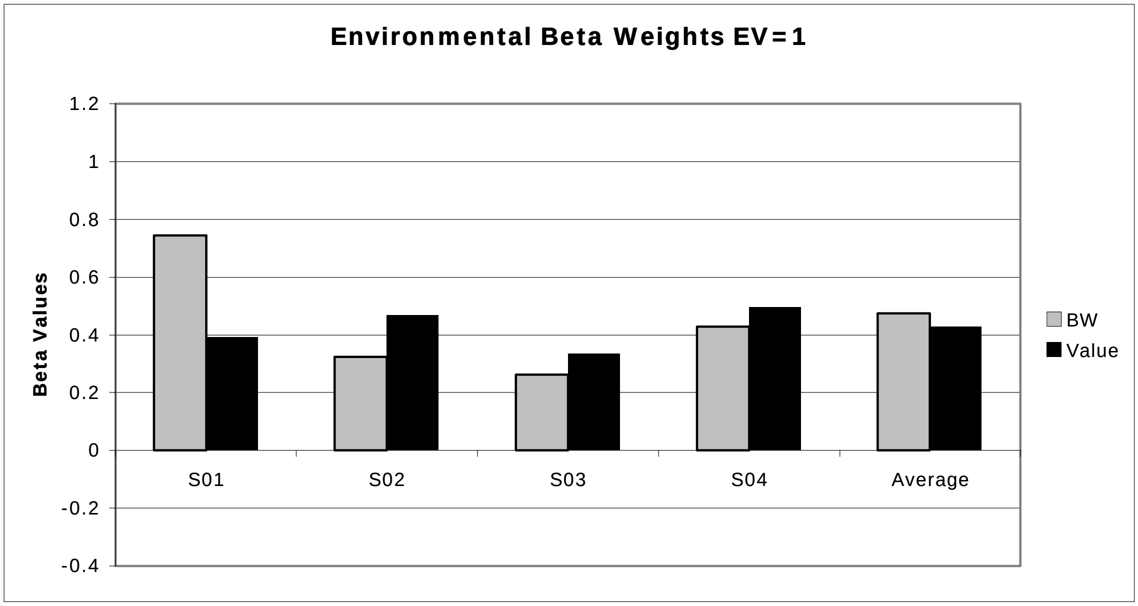


Figure 17. Environmental cue weights EV=1.

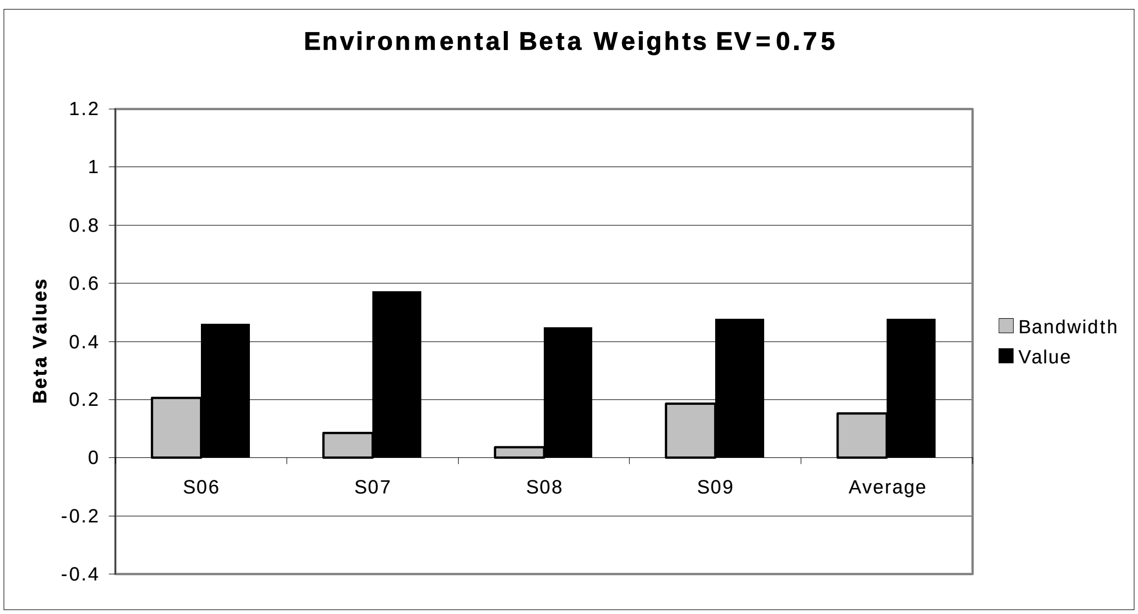


Figure 18. Environmental cue weights EV=0.75.

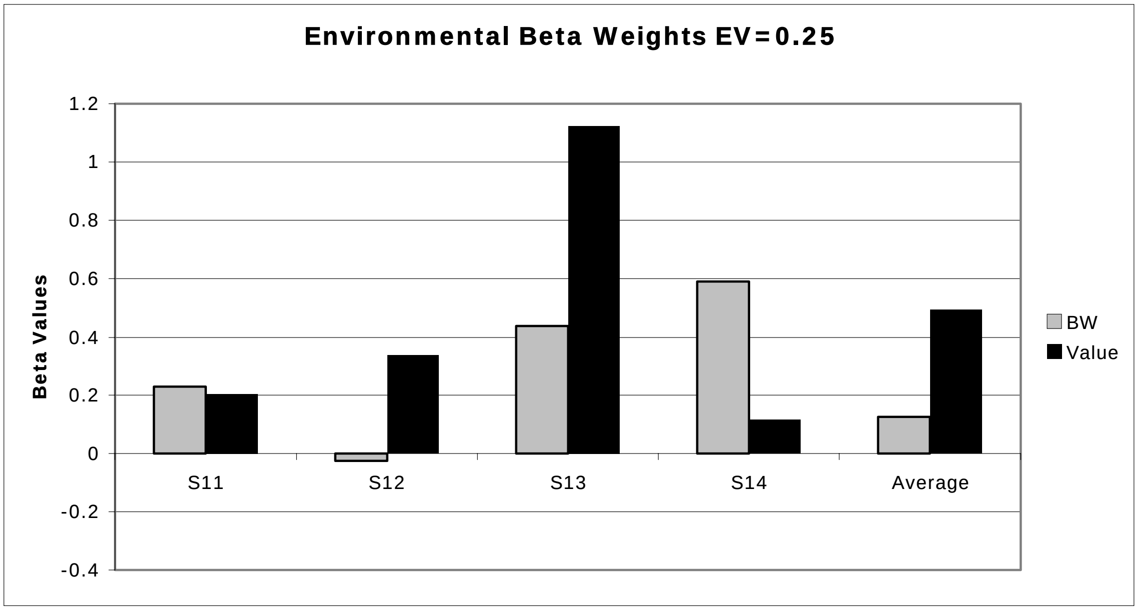


Figure 19. Environmental cue weights EV=0.25.

As expected, at the overall level per EV group, the optimal beta weights for the BW cue went down as EV went down, while the weights for the value cues remained relatively constant across conditions. Somewhat surprisingly, there was a large difference in the beta weights for the BW cue between the EV=1 condition (0.48) and the EV=0.75 condition (0.15), but very little difference between the EV=0.75 and EV=0.25 (0.15 versus 0.13 respectively). This finding begins to shed light on a possible explanation of the performance results that found that there was very little difference between the EV=0.75 and EV=0.25 conditions. Though, the environmental model cannot explain participant performance, it can be used to show that that there was very little difference with respect to the *optimal* cue weighting strategies between the two EV<1 conditions. These findings can then be used in light of the Lens Modeling to understand how the different EV conditions were affected by the different (or similar) environments.

Another surprising finding illustrated in the environmental modeling is how different the environments were *within* EV conditions. Notice, especially in EV=0.25, how different the optimal beta values were across participants. The optimal weights for the BW cue ranged between -0.03 and 0.59. In fact, S13 and S14 from the EV=0.25 condition were expected to weight BW more heavily than S02 and S03 in EV=1. This was unexpected because the goal of the study was to manipulate the correlation between BW and alarm rate, which means that the beta weight for the BW cue *should* be lower as EV is reduced.

It appears that the differences between participants within a given EV condition could be explained through the design of the study. Since value played a role in calculating the optimal time on each gauge (alarm rate x value), different combinations of value combined with alarm rate for a particular gauge could make a difference in the optimal beta weights for the BW and value cues for that participant. This effect is illustrated in the following example: In a given trial (EV=0.25), S12 BW=4 was coupled with value=2 for a gauge, while the BW=4 was coupled with a value of 8 for S14. Alarm rate for this gauge, for both participants, was 0.08 alarms per second. This means that, even though the BW was the same, S14 should have put much more emphasis on this gauge than S12 (higher weight on BW cue). Although the optimal beta weights are based on the combination of all four gauges rather than a single gauge, it can be seen that different combinations of BW and value can differently affect the cue weightings for both cues.

5.4.3 Early Trials (Trials 1 – 10)

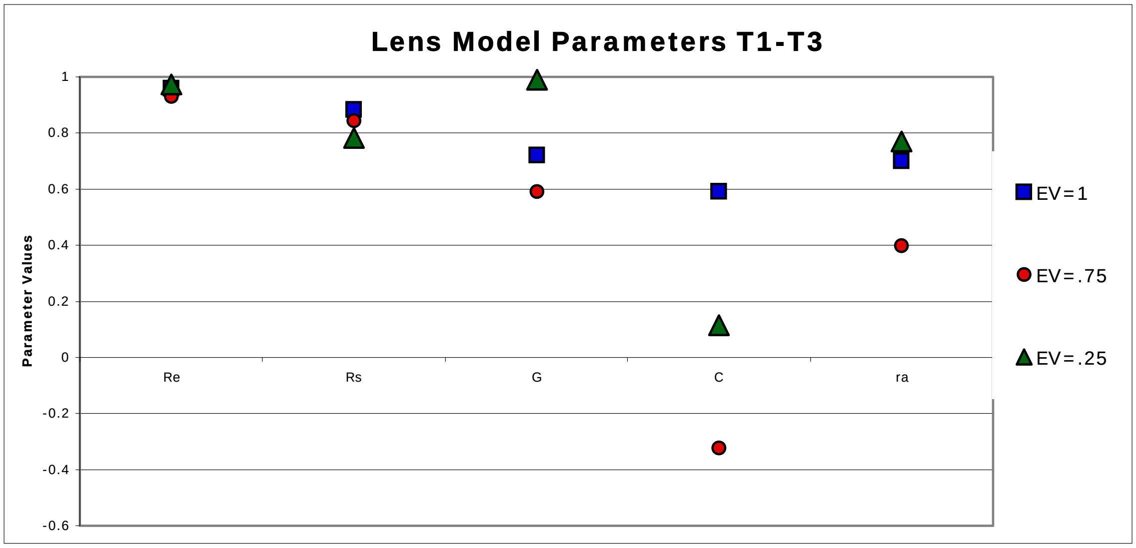
The early trials were composed of trials one through ten. Eye movement data were collected and analyzed in sets of three trials (T1-T3, T4-T6, and T8-T10). Recall that it was necessary to group trials by sets of three to have enough data points to obtain significant models. Although we modeled data per individual and across EV conditions for each set of three early trials, little insight into learning was gained from Lens Modeling at this fine grain of detail. Therefore, only a brief overview of the modeling results related to individual participants will be presented in this section as well as a high-level overview of the modeling results for each of three sets of early trials. The detailed graphs for each set of early trials are provided in Appendix C.

First, modeling individual participants over each set of three trials revealed that there were large individual differences across participants within and between EV conditions. These individual differences were primarily observed in the knowledge component (G), and correspondingly in the achievement (r_a) component. Recall that G measures the extent to which the beta values for the observed BW and value cues are in proportion to the optimal beta values, while r_a is the correlation between the optimal and observed percent time on each gauge. Both environmental predictability (R_e) and consistency (R_s) were similarly high (close to 1) across all participants except where noted. Although C varied across participants and EV conditions, since R_e and R_s were close to one, C did not largely contribute to achievement. Recall that achievement is measured as a combination of C, G, R_e and R_s and the contribution of C to achievement is moderated by $\sqrt{1-R_e^2}$ and $\sqrt{1-R_s^2}$. Since R_e and R_s were relatively high, G was the primary factor driving achievement across participants.

Figures 20, 21, and 22 present the mean Lens Model Parameters and beta values for each EV condition across T1-T3, T4-T6, and T8-T10 respectively. Figure 20 presents the Lens Model parameters and cue weights for T1-T3. For the first set of trials (T1-T3), only 7 of 12 participants could be successfully modeled ($p<0.05$), so the figures should be interpreted with some caution. Environmental predictability (R_e) and consistency (R_s) were similar across all EV conditions. Achievement (r_a), which was calculated as the correlation between the optimal and actual percent time on each gauge was slightly higher for the EV=0.25 group than the EV=1 group and much higher than the EV=0.75 group. The high level of achievement and linear knowledge (G) for the EV=0.25 condition may have been, at least partially, due to the participants who could be modeled at this stage.

Since it is expected that the most salient cue (BW) would drive sampling at this early stage, the environment that most heavily weights BW as a cue should have higher values for both G and r_a. Recall from the environmental models in the EV=0.25 condition (see Figure 19) that that S13 and S14, to perform optimally, should weight the BW cue higher than some participants in the

EV=1 condition. Therefore, what originally looks like an unexpected result may actually be in line with the expectation that BW is driving sampling.



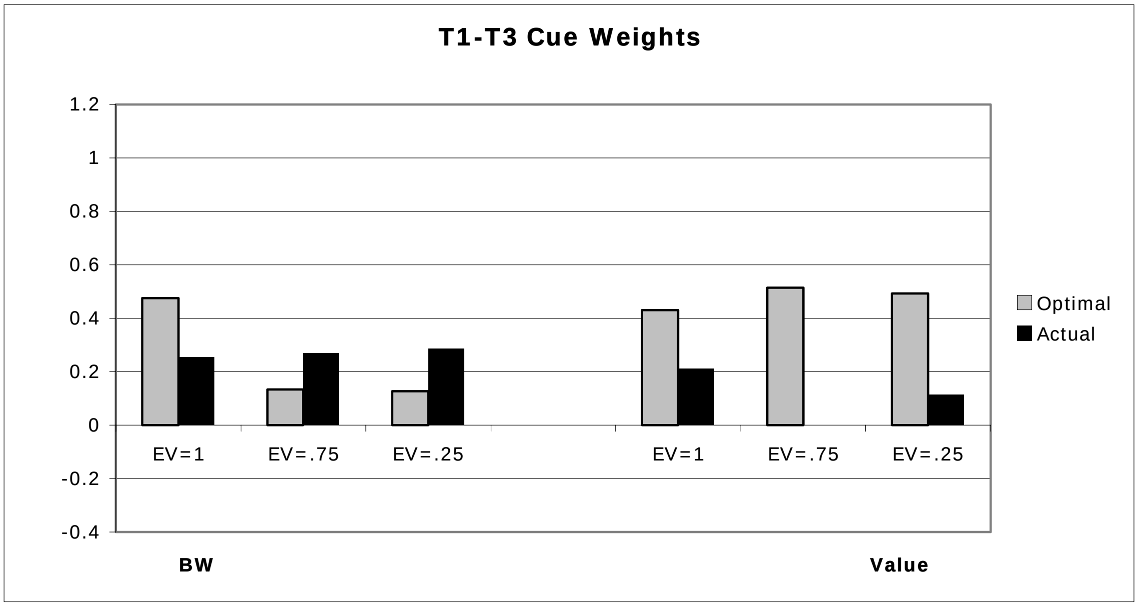
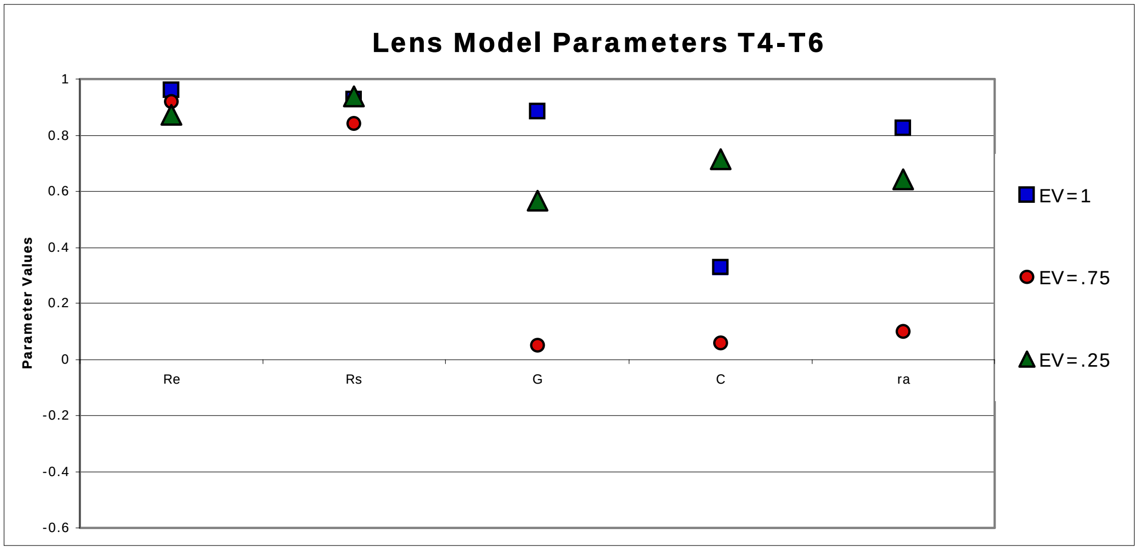


Figure 20. Lens Model parameters and cue weights for T1-T3 per EV condition.

Note that the averages are based on participants who could be reliably modeled for each condition. In EV=1, S02 and S04 composed the average. In EV=0.75, S07, S08, and S09 were modeled. In EV=0.25, S13 and S14 were used in modeling. The value cue utilization for EV=0.75 was modeled, but it was close to 0.



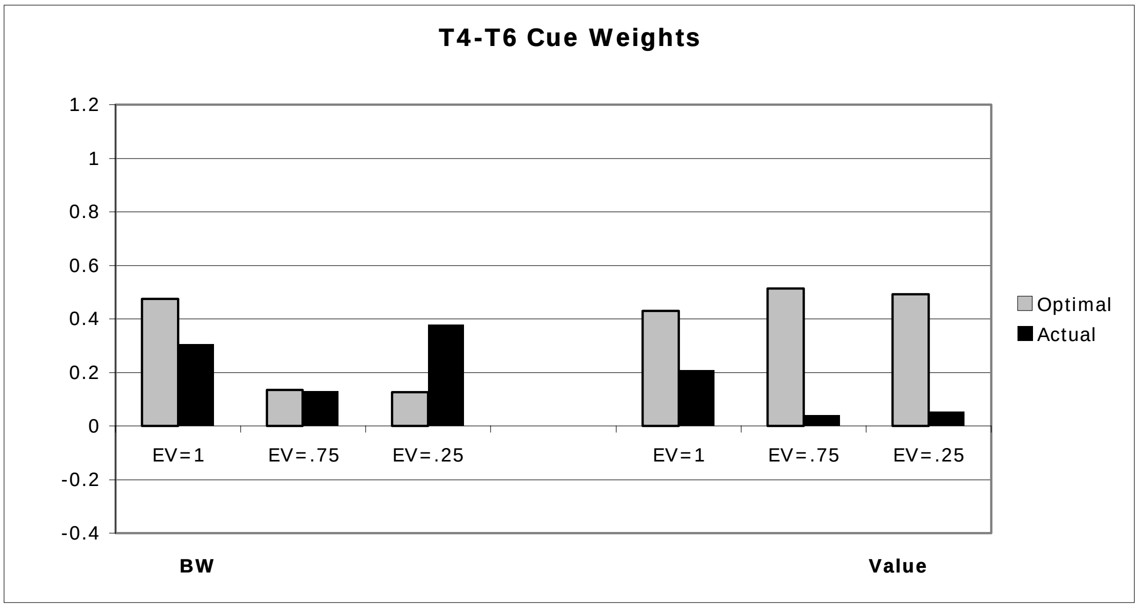
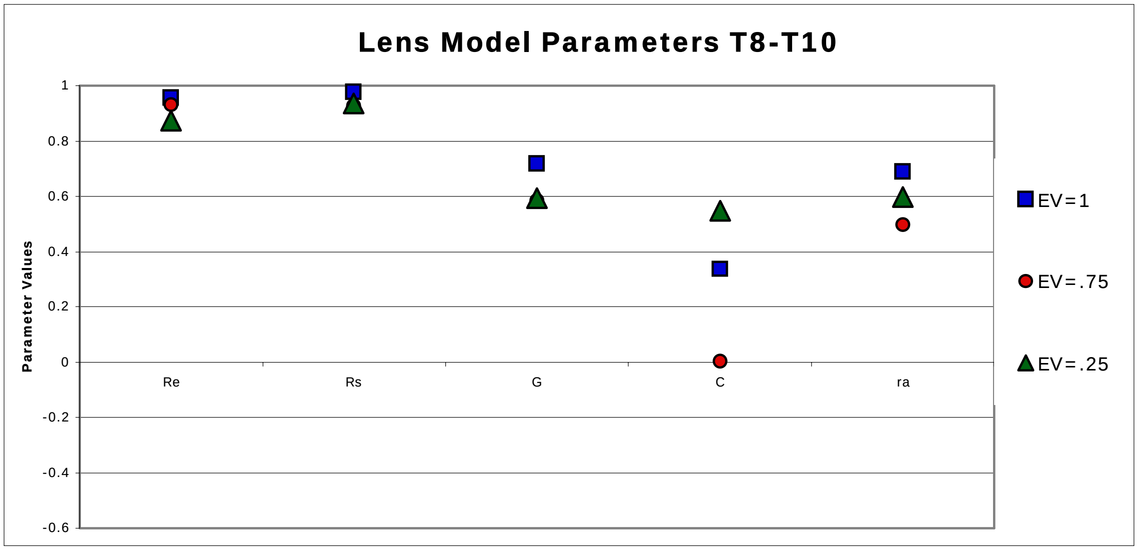


Figure 21. Lens Model parameters and cue weights for T4-T6 averaged per EV condition.

Note that the averages are based on participants who could be reliably modeled for each condition. In EV=1, S01, S02 and S04 composed the average. No eye movement data was collected for S03, therefore no optimal or actual data were included in the analysis. In EV=0.75, S06, S07, S08 were modeled. No eye movement data were collected for S09, therefore no optimal or actual data were included in the analysis. In EV=0.25, S12, S13 and S14 were used in modeling.



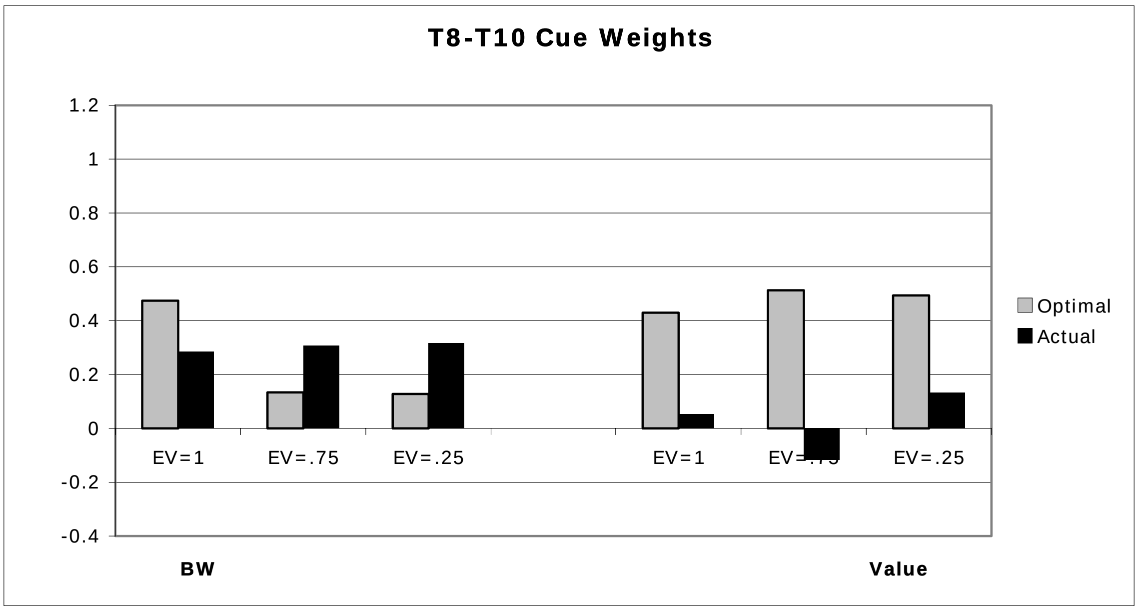


Figure 22. Lens Model parameters and cue weights for T8-T10 averaged per EV condition.

Note that the averages are based on participants who could be reliably modeled for each condition. In EV=1, S01, S02, S03 and S04 composed the average. In EV=0.75, S07, S08, and S09 were modeled. In EV=0.25, S12, S13 and S14 had significant models.

Figure 21 shows the Lens Model parameters and cue weights for T4-T6. At this point, 11 out of 12 participants could be reliably modeled. No eye movement data were collected for S03 and S09, while S11 could not be reliably modeled at $p<0.05$. The results from the Lens Model analysis shows a difference is beginning to emerge, separating the EV conditions. This figure shows that linear knowledge, achievement, and unmodeled knowledge are widely varied depending on EV condition. For both G and r_a were highest for EV=1 (0.88 and 0.82 respectively) followed closely by EV=0.25 (G=0.56 and r_a=0.64). Participants in the EV=0.75 condition, on average, were not knowledgeable and had correspondingly low achievement (G=0.05 and r_a=0.10). Since both R_e and R_s were similar across conditions (though R_s was somewhat lower for EV=0.75), C did not contribute to achievement. This means that the low achievement in the EV=0.75 condition was likely due to a low value of G rather than C.

The final set of early trials coded was T8-T10, presented in Figure 22. The only participants that did not have significant models at this point were S06 (EV=0.75) and S11 (EV=0.25). The results from the Lens Model analysis also show that the linear knowledge and achievement parameters were beginning to converge by these late-early trials. Note that the only parameter that is different across EV conditions is the unmodeled knowledge parameter. As suggested earlier it would be advantageous to performance for participants to have non-linear knowledge, since the optimal percent time on gauge (and performance) is based on a multiplicative rather than additive function. Though C is lower for EV=0.75, the parameter contributes little to the final achievement.

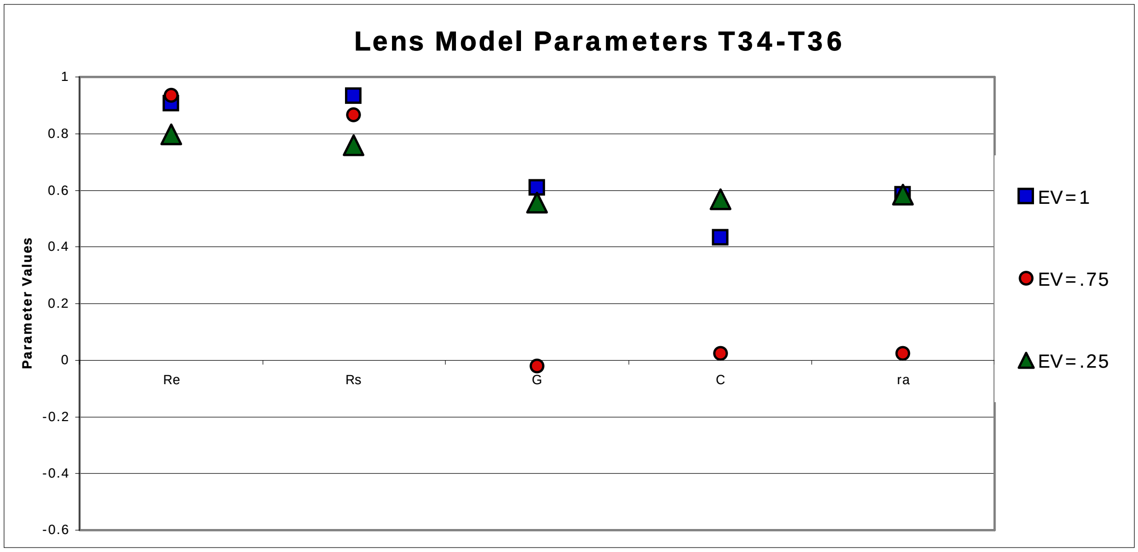
A comparison of the average beta values for the BW and value cues, shown in Figure 22, reveals that the BW cue was weighted the same across all EV conditions for the participant models. This suggests that BW is still a salient cue that is driving sampling irrespective of how much the cue should be weighted. Value still wasn't highly weighted by either the EV=1 or EV=0.25 and was negatively weighted in the EV=0.75 condition.

5.4.4 Late Trials (Trials 34 – 36)

The final three pre-transfer trails (T34-T36) were modeled in order to investigate adaptation late in the study. This section provides both an overview and a detailed analysis of the Lens Model Parameters and beta values for each EV condition. In addition to this analysis, in order to paint a clearer picture of the effects of the EV manipulation on visual attention allocation, a comparison of the optimal versus the observed fixations is presented (i.e. achievement). The optimal versus observed comparison allows a more detailed breakdown of the Lens Model using the skill score decomposition (see Cooksey, 1996; Stewart and Lusk 1994; Kirlik and Strauss in press).

Figure 23 presents an overview of the Lens Model parameters and cue weights for T34 through T36. By this time, all participants could be reliably modeled at $p<.05$. The most interesting and striking difference between the EV conditions is that the visual scanning strategies of those in the EV=0.75 condition had *no* linear knowledge, non-linear knowledge, or achievement at this late stage, while both EV=1 and EV=0.25 were both similarly adapted to their respective environments, as indicated by their knowledge (G) and achievement (r_a) levels. This clearly lends support to the previous suggestion that, although performance scores were approximately the same across all EV conditions, those in the EV=0.75 condition were not as adapted to the task as the other EV conditions.

However, as shown in Figure 23, there is little difference between the beta weights for the BW cue across EV conditions, rather the value cue was where the primary difference was observed. As suggested previously, one might expect that participants would first adapt to the more salient BW cue before beginning to adapt to the value cue. If the EV=0.75 group was indeed in the more



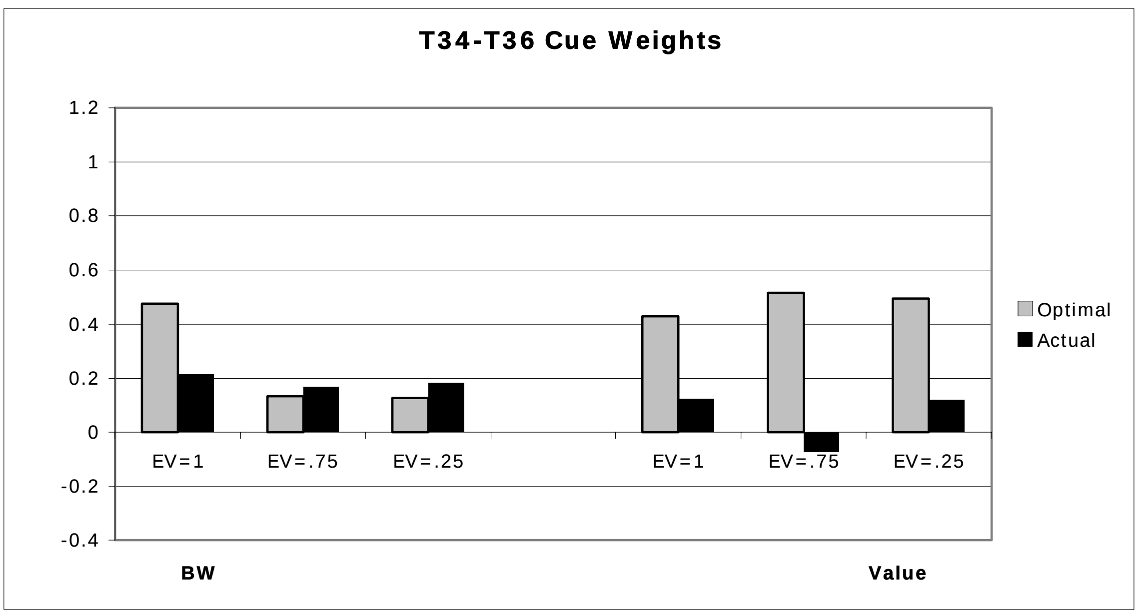


Figure 23. Lens Model parameters and cue weights for T34-T36 averaged per EV condition.

Note that the averages are based on participants who could be reliably modeled for each condition. All participants produced significant models for the final set of trials.

difficult environment, then it might be that this group had just begun to understand how to weight BW and was not yet incorporating value into their scanning. The EV=1 and EV=0.25 conditions may have, as optimally as possible, adapted to the BW cue (perhaps early) and had also begun to adapt to the less salient value cue.

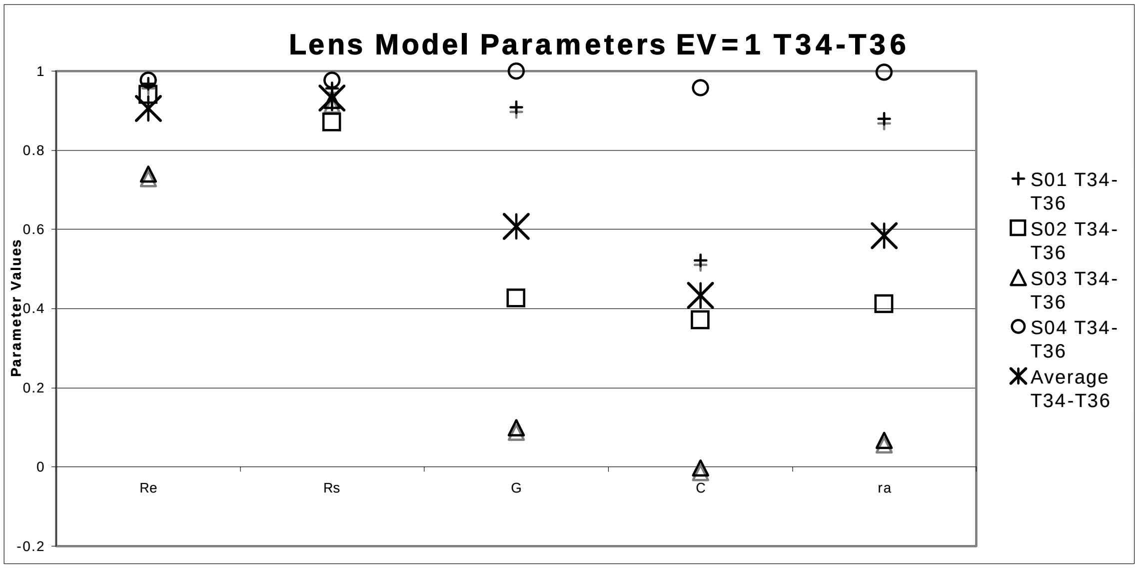
Also note that the EV=1 and EV=0.25 groups were both making use of non-linear knowledge, while the EV=0.75 group was not. It could be that adaptation in this task requires use of unmodeled knowledge (C). Groups that take most advantage of this non-linear knowledge are those that are more optimally adapted than those who do not. However, since R_e and R_s were relatively high, C did not contribute very much to achievement in most cases (except see the EV=0.25 condition).

Figure 24 illustrates the Lens model parameters and beta values for the EV=1 condition over the final set of trials. Notice that the values for G, C, and r_a range from near 0 to 1. Though C is close to 1 for S04, C contributes less than 5% to the total achievement score. Recall that achievement is measured as the sum of the linear component (R_e x R_s x G) and the unmodeled component ($C\sqrt{1-R_e^2}\sqrt{1-R_s^2}$). However, it might be argued that since performance (measured at total score) peaked out at about 85%, one way to differentiate performance would be through non-linear knowledge. Total score is measured as value multiplied by alarm rate, so it might be that the better performers differentiate themselves from the average by their use of non-linear knowledge.

The cue weights indicate that S04 was the only participant to weight the value cue, so the average cue weight for value was due to this participant. BW was similarly weighted across all participants in this condition. It appears that the top performer in this group (S03) was most appropriately weighting the BW cue.

EV=1 T34-T36

A more intuitive way of visualizing achievement (r_a) is shown in Figure 25 as the predicted percent time on a gauge versus the observed time on a gauge. Achievement is the correlation between optimal environmental gaze duration and the participants gaze duration for the corresponding gauge. There was some earlier discussion regarding the observed discrepancies between performance and achievement, where it was suggested that some additional information might be gleaned from a more detailed analysis of achievement. Recall that the Lens Model equation, used for this study, measured achievement as the correlation (shape) between optimal and observed scanning. This left open the possibility that some differences between achievement, as calculated by the Lens Model, and observed performance, as calculated by the total score, might be due to other factors like regression (range distribution) or base rate (over- or under-estimation) bias. These biases have been used in judgment analysis to explain observed differences between “potential” performance and “actual” performance (Cooksey, 1996).



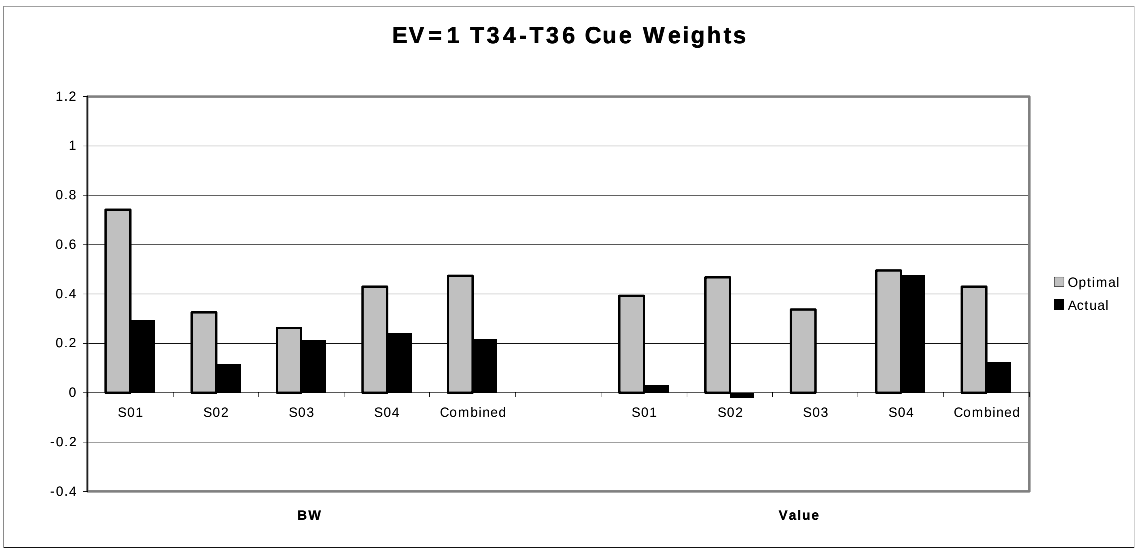


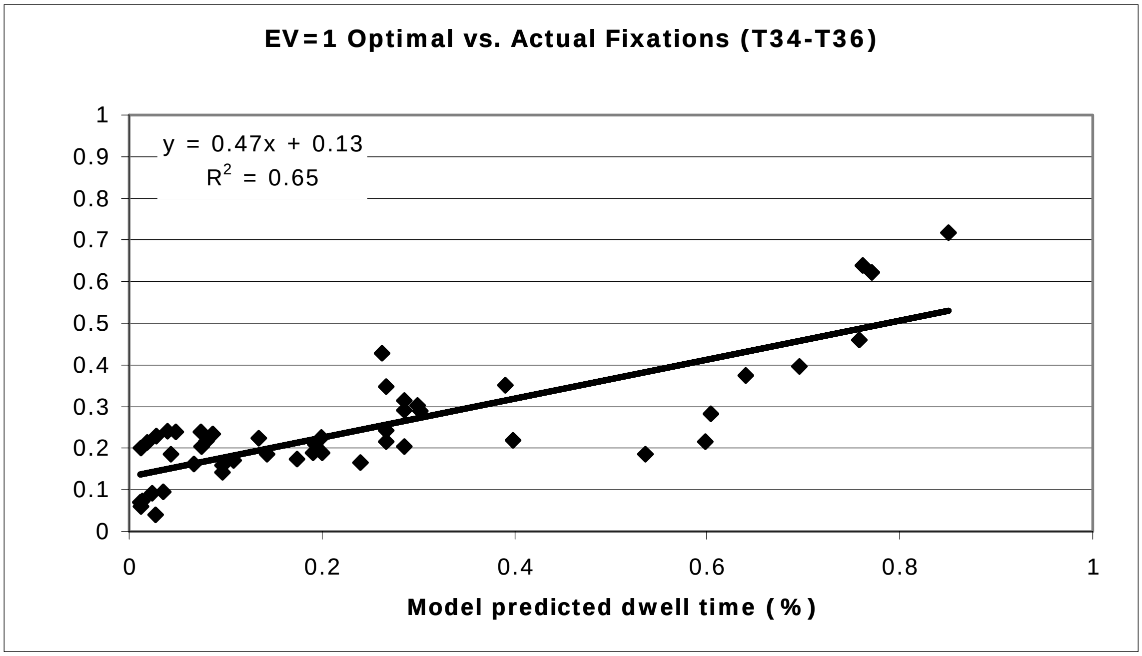
Figure 24. Lens Model parameters and cue weights for T34-T36 EV=1.

As expected, based on the achievement score, the optimal and actual percent dwell time for EV=1 condition were highly correlated. However, the slope is somewhat flattened (m=0.47) with a base rate bias (b=0.13). The individual graphs illustrating optimal versus actual percent time on each gauge show striking individual differences. These values range from a slope of 0.75 and a y-intercept of 0.06 to a slope of 0.10 and an intercept of 0.22. Based on these results, if the skill score is correlated with performance, it would be expected that S02 and S03 would be the lowest performers in the group.

Although this modeling was intended to address scanning, it is of interest to compare the results from the modeling with the performance measures because it should give us some insight into how the modeling compares with performance. It appears that sometimes the skill scores are relatively predictive of performance. For example, S02 above had the lowest total score (%) in the condition over the final three trials and had a correspondingly low achievement score.

However, S03 had a similarly low skill score, but he was the top performer in this condition. Therefore, either r_a and skill score do not correspond with performance or there was some other factor influencing performance that was not captured by this modeling technique.

The remaining factor that could help explain the differences between performance and achievement or skill score that is not being captured by this modeling technique was the distinction between fixation duration and percent of time fixating on each gauge. The Lens Modeling compared the observed and predicted percent of time spent fixating on each gauge, which combines fixation frequency and fixation duration into a single term. Therefore, the same percent time on a gauge could be obtained from making many short fixations or few long ones. It could be assumed that many short fixations imply less need for selective sampling. For example, if an operator made many short fixations (e.g. 1/3 sec/gauge), all four gauges could be observed in a little over a second. Operators who had short fixation durations could almost sample all the gauges without considering BW or value and still catch most of the alarms within the time limit. This means that the observed percent time on each gauge could be around 25% for all gauges with a correspondingly high performance. However, if the dwell duration increased to 1/2 sec/gauge or longer, then it would take 2 or more seconds to sample all the gauges. The longer dwell duration would imply that these operators would have to be more “optimal” in their sampling strategy to achieve the same performance. Thus, the same alarm detection rate could be observed through high or low achievement depending on fixation duration. Fixation duration as a potential mediator of performance and achievement will be examined in the following section (Section 5.5).



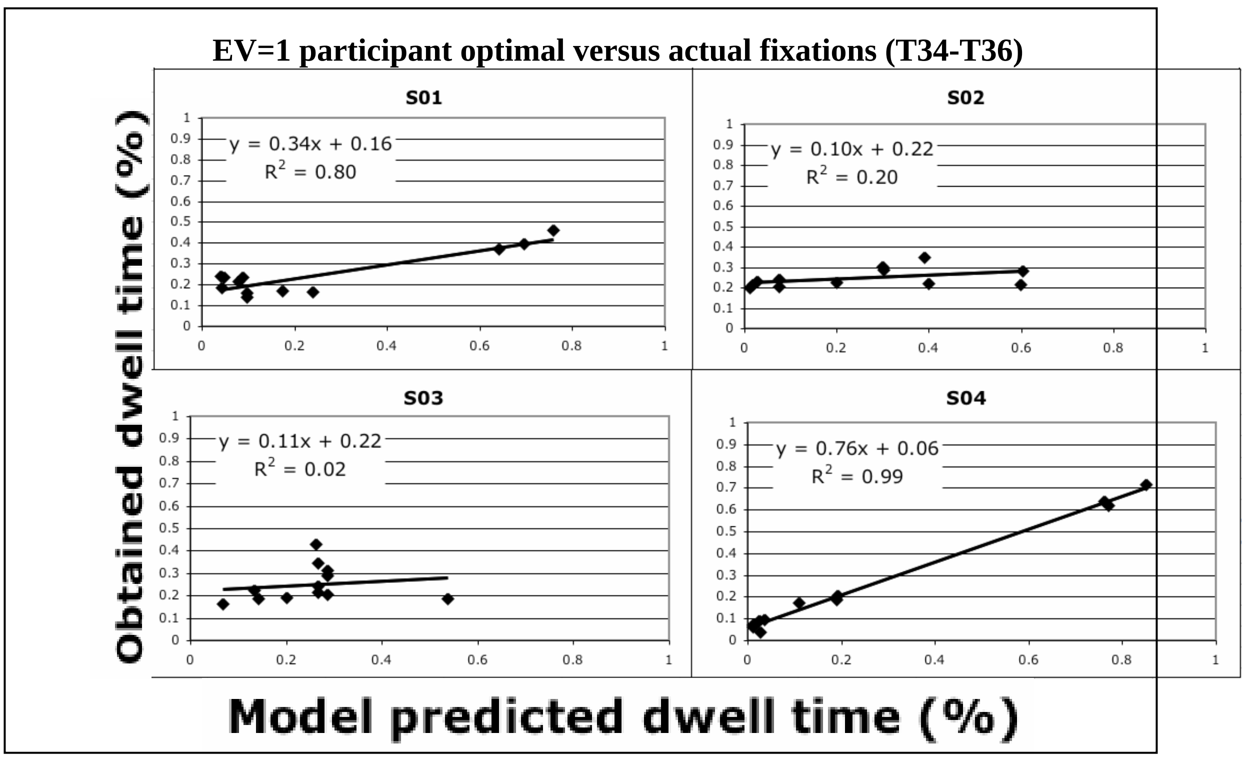


Figure 25. EV=1 condition. Model predicted dwell time (%) by obtained dwell time (%). The top graph includes all data points across participants S01-S04 and trials (T34-T36), while the bottom four graphs are individual dwell times over each of the final three trials (T34-T36).

<u>EV=0.75 T34-T36</u>

Again, like all the other trials and EV conditions Figure 26 shows that G drives achievement in the EV=0.75 condition for the final set of pre-transfer trials. In the EV=0.75 condition the G values, on average, were lower than the EV=1 condition. In this condition, S09 was the top performer and had the highest achievement score. Though S09 was the top performer in this condition, his achievement score was lower than all but the lowest achievement score in the EV=1 condition (but this participant was the top performer).

The cue weights from Figure 26 show that most participants were over-weighting the BW cue and not weighting or negatively weighting the value cue. Like the EV=1 condition, the top performer in this condition (S09) was most appropriately weighting the BW cue. The value cue was highly negatively weighted for this participant.

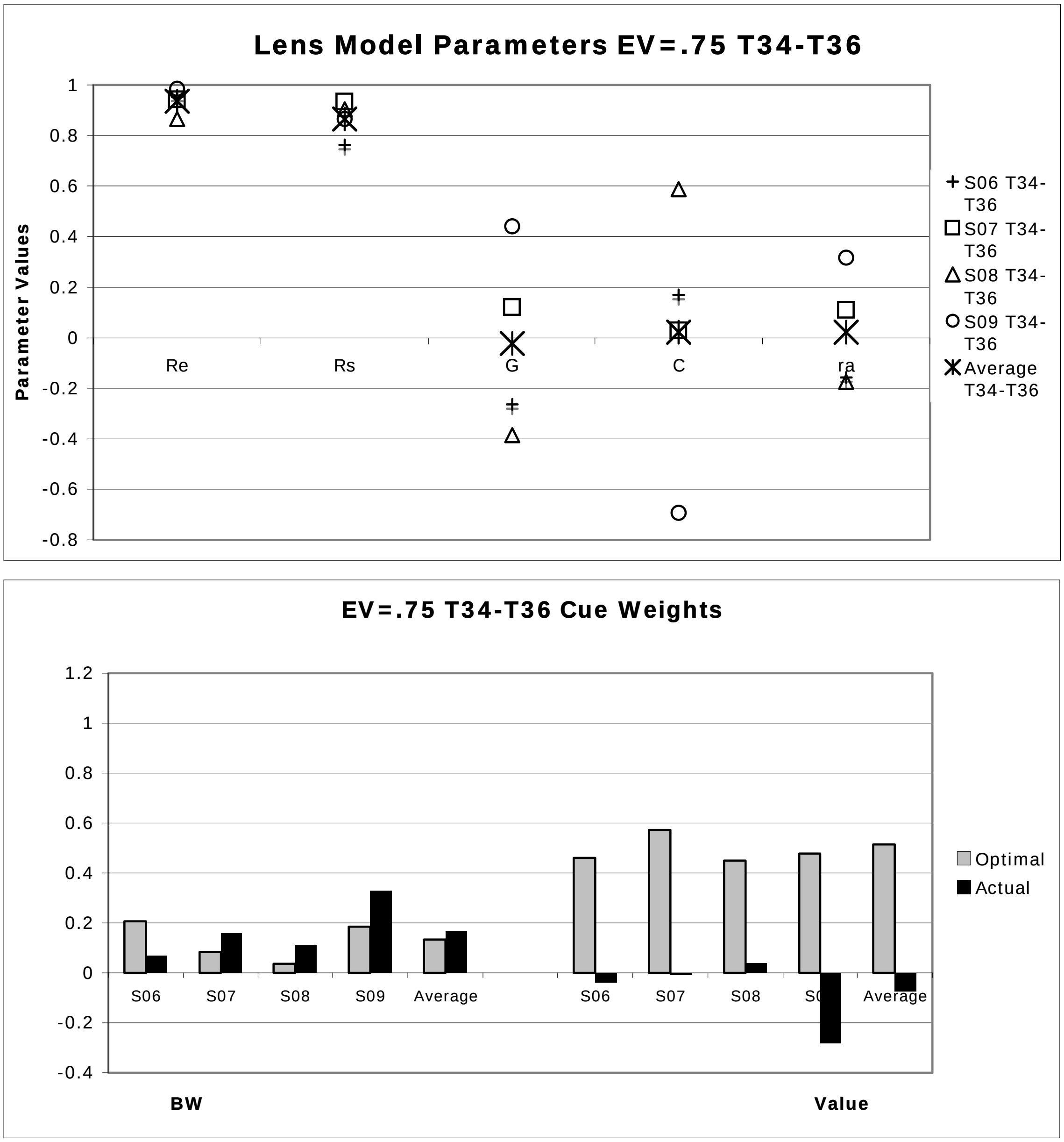
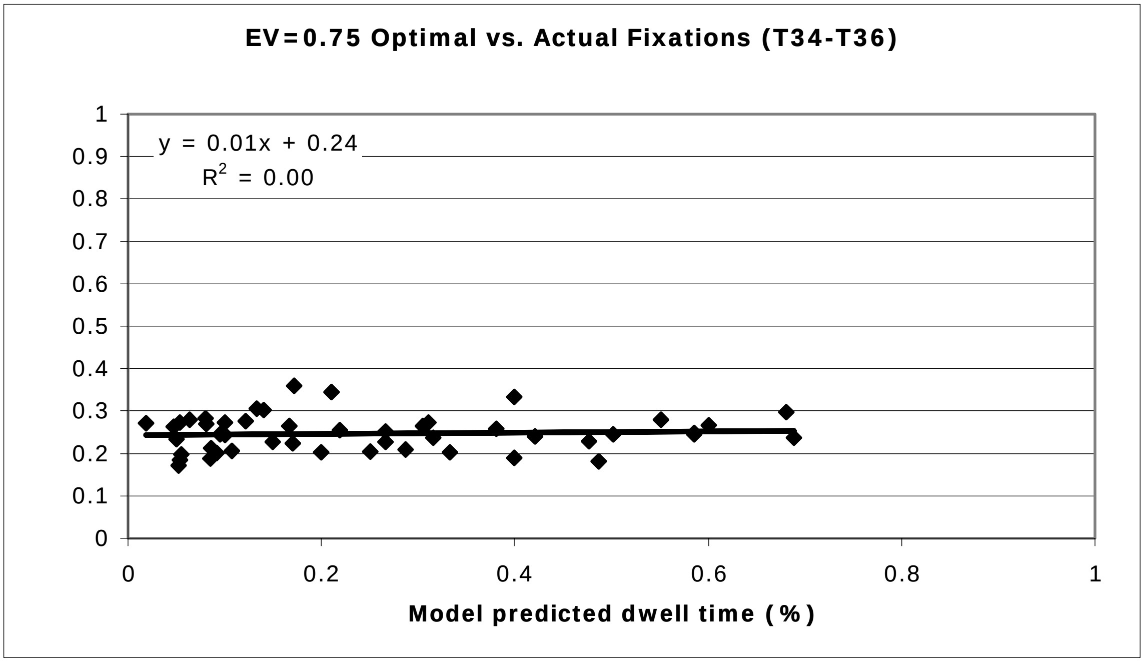


Figure 26. Lens Model parameters and cue weights for T34-T36 EV=0.75.

The optimal versus actual percent of time fixated on each gauge is shown in Figure 27. Figure 27 clearly shows that the participants in this condition had an almost uniformly flat search slope (m ranged from -0.05 to +0.07). This flat slope provides clear support to the growing body of evidence that the visual scanning strategies of participants in this group were not well-adapted to their environment. Although it was possible that some of the participants were compensating for the non-optimal scanning pattern, as explained previously, it is unlikely that all the participants in this group were “super-scanners”.



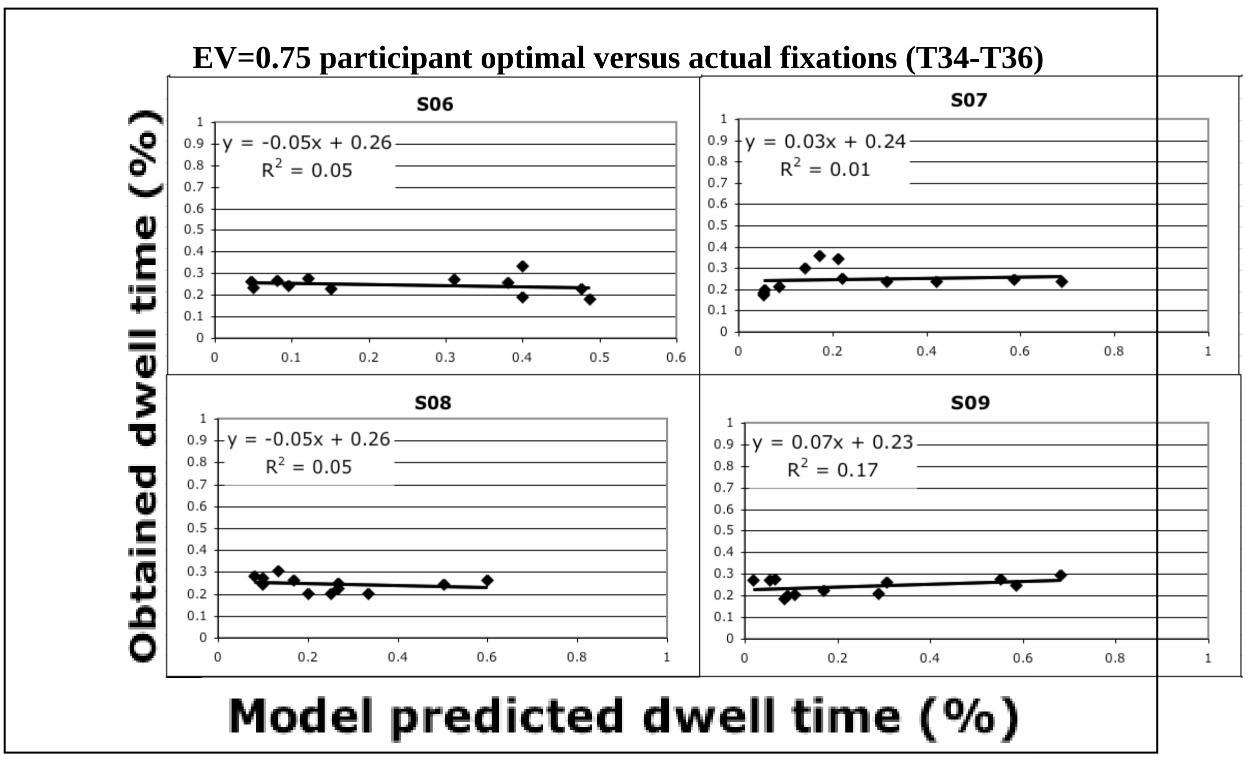
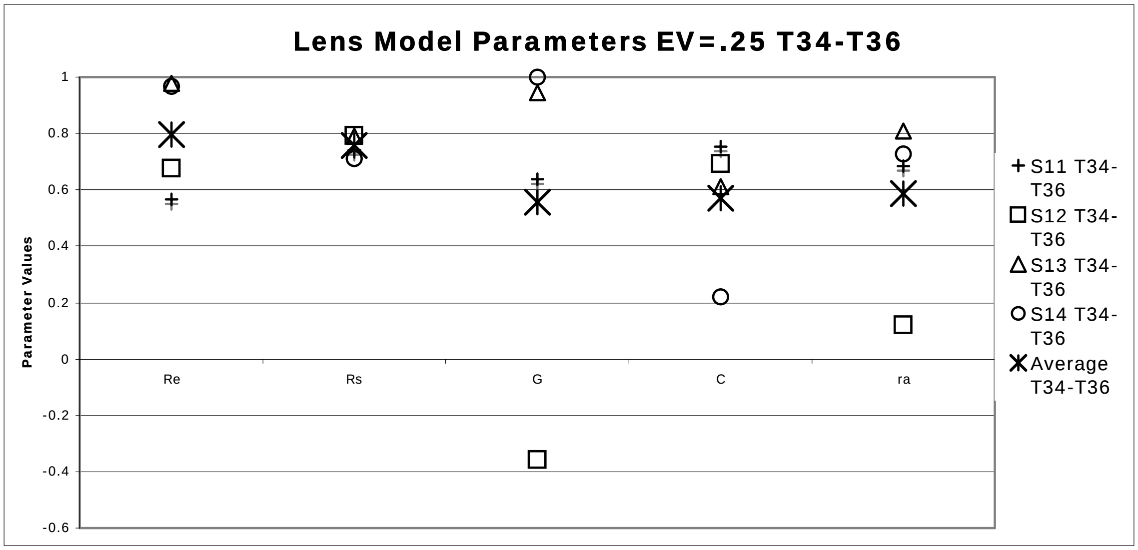


Figure 27. EV=0.75 condition. Model predicted dwell time (%) by obtained dwell time (%).The top graph includes all data points across participants S06-S09 and trials (T34-T36), while the bottom four graphs are individual dwell times over each of the final three trials (T34-T36).

<u>EV=0.25 T34-T36</u>

Finally, Figure 28 shows the Lens Model parameters for the final trials in the EV=0.25 condition. This figure illustrates that there was a difference in the environments between S11 and S12 and S13 and S14, but there was no difference in consistency across participants. The unmodeled knowledge component was a much bigger contributor to overall achievement for S11 and S12 than for S13 and S14 (.55 and .45 versus .13 and .18 respectively). This is interesting finding, because it could suggest that the participants in the less informative environment (S13 and S14) *had* to make use of non-linear knowledge to adequately adapt to the task. The top performer in the group (S11) made the most use of non-linear knowledge.

Another interesting finding was that S11 was the best adapted to environmental weighting of the BW cue. This means that for all three conditions, the top performer in terms of total score was also the best adapted in terms of the BW cue weighting. The value cue was not predictive of performance.



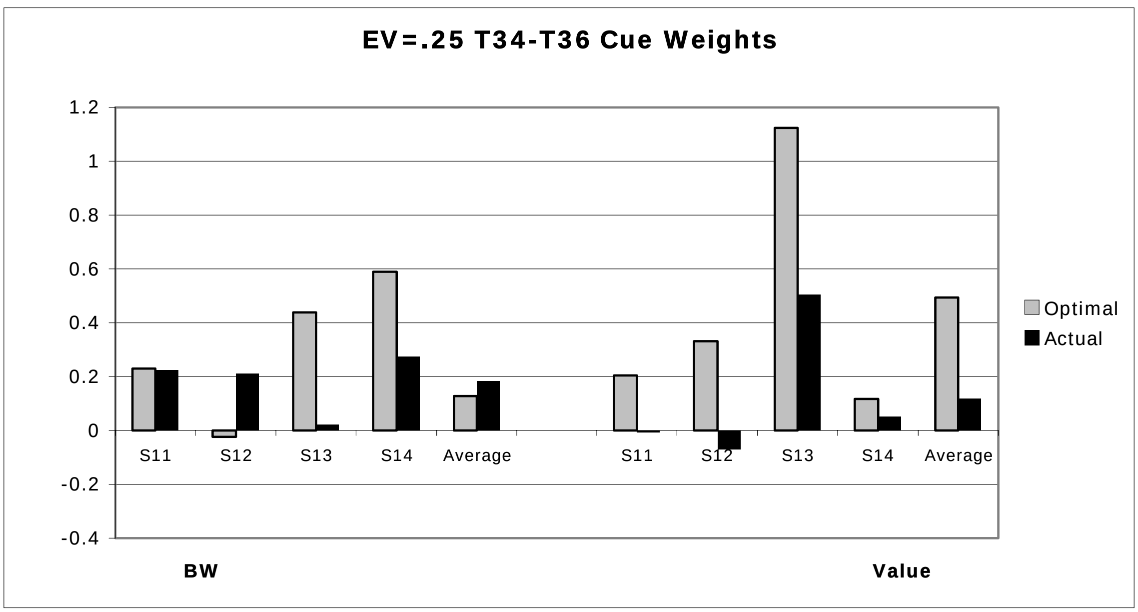
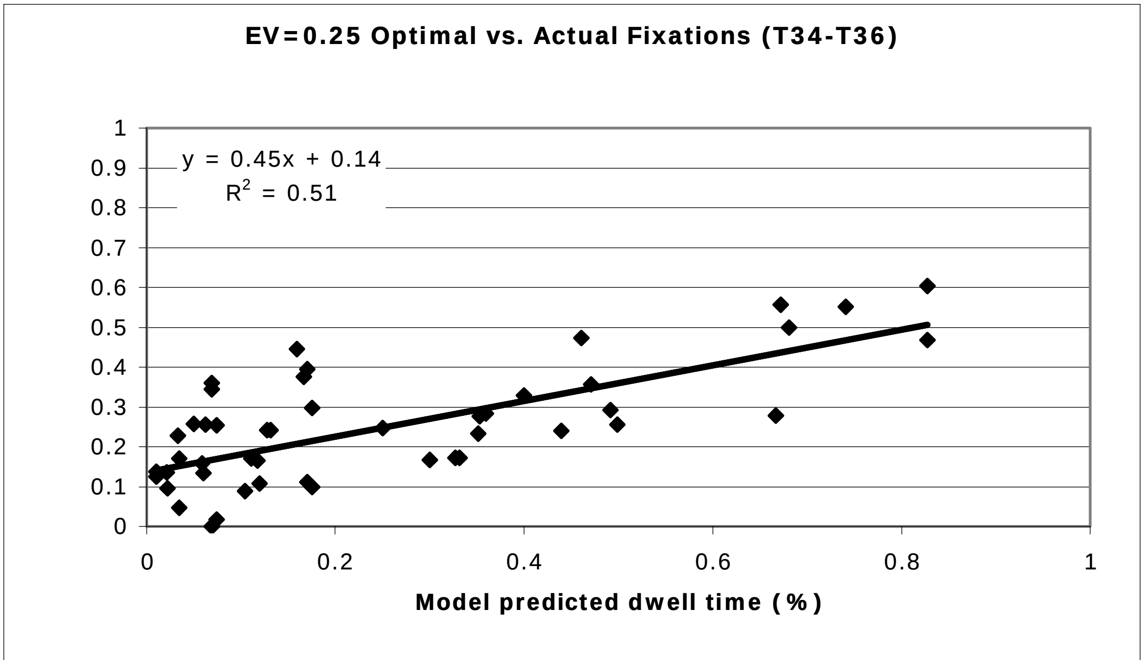


Figure 28. Lens Model parameters and cue weights for T34-T36 EV=0.25.

Figure 29 shows the optimal versus actual percent time on each gauge for the EV=0.25 condition. Like the EV=1 condition, the slope in this condition was positive. The slope for the EV=0.25 condition was flatter, and the observation points were more scattered than the EV=1 condition. This suggests that the participants in the EV=0.25 condition were more adapted to their environments than the EV=0.75 condition, but less adapted than the EV=1 condition.



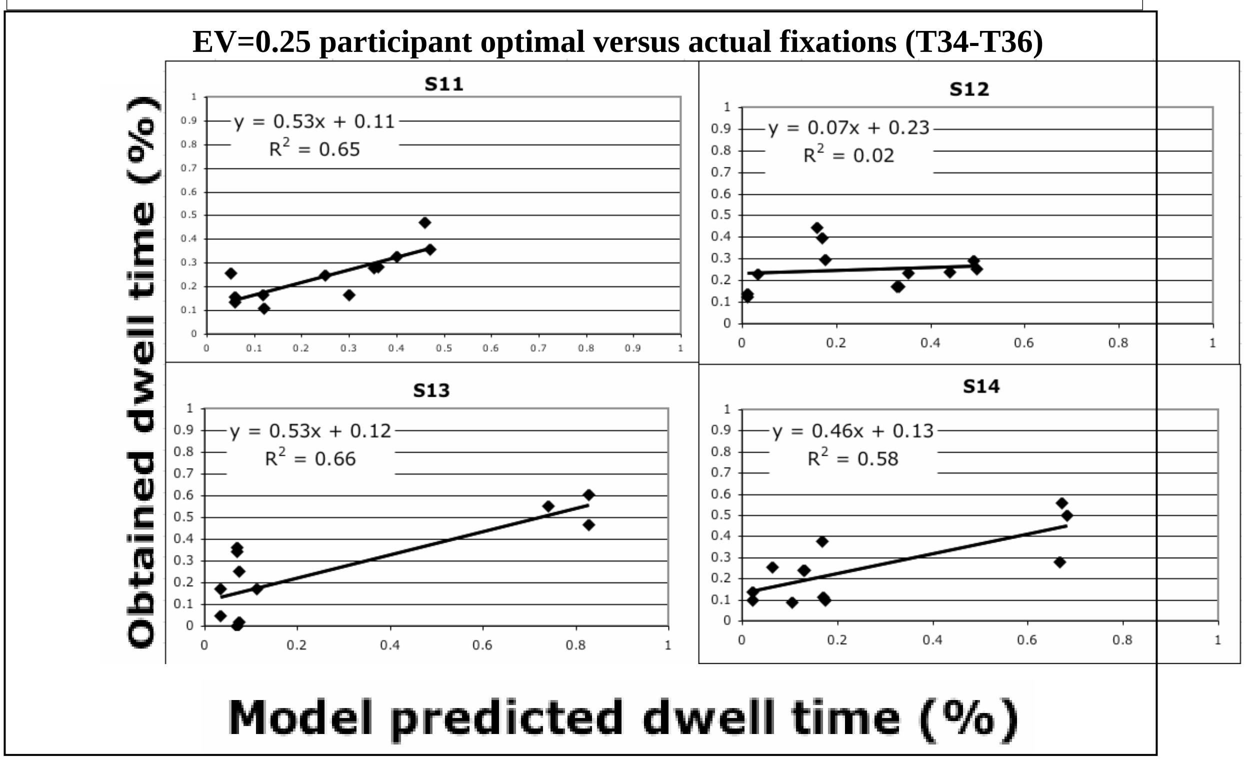


Figure 29. EV=0.25 condition. Model predicted dwell time (%) by obtained dwell time (%).The top graph includes all data points across participants S11-S14 and trials (T34-T36), while the bottom four graphs are individual dwell times over each of the final three trials (T34-T36).

5.4.5 Comparison Between Fixation Frequency and G

As discussed earlier, a possible reason for the apparent discrepancy between performance and the adaptivity of visual scanning strategies across participants could be explained by looking more closely at fixation duration. The addition of shorter fixation duration (more fixations per unit time) as a potential contributing factor complicates the analysis somewhat because the implications are not clear. On one hand, shorter fixations mean that a gauge can be sampled more often. On the other hand, longer fixation durations could imply that more information can be extracted from the environment per fixation. Although shorter fixation durations imply that the operator is not required to scan more “optimally” to achieve high performance, operators could still employ an optimal scanning strategy. This means “super fixators” can either employ an optimal or non-optimal scanning strategy to achieve good performance. However, participants who have long fixation durations must employ an optimal scanning pattern to achieve a high level of performance. If the slow fixators are not optimal in their scanning patterns, then we should observe lower performance.

To determine if participants who had a higher number of fixations per trial were indeed less optimal in their scanning pattern, G (knowledge) was compared with the number of fixations per trial, and shown in Figure 30 below. If operators with fewer fixations needed to be more adapted to the BW and value cues, it would be expected that fixation frequency would be inversely proportional to G (correct weighting of the cues). This analysis was conducted for all participants over T34-T36. Each participant’s fixation frequency was recorded individually for each of the three trials and compared with the G value for each participant. Since G was calculated over T34-T36, the same G value was recorded for each of the three trials per participant. A significant negative correlation was found between G and fixation frequency ($t_{34} = -2.8$; $p<0.05$). This suggests that participants who made more fixations per trial were not as optimal in their scanning strategies, but it also implies that these participants did not *have* to be more optimal to perform well with respect to the task criterion because they could sample the gauges more often than those who did not make as many fixations.

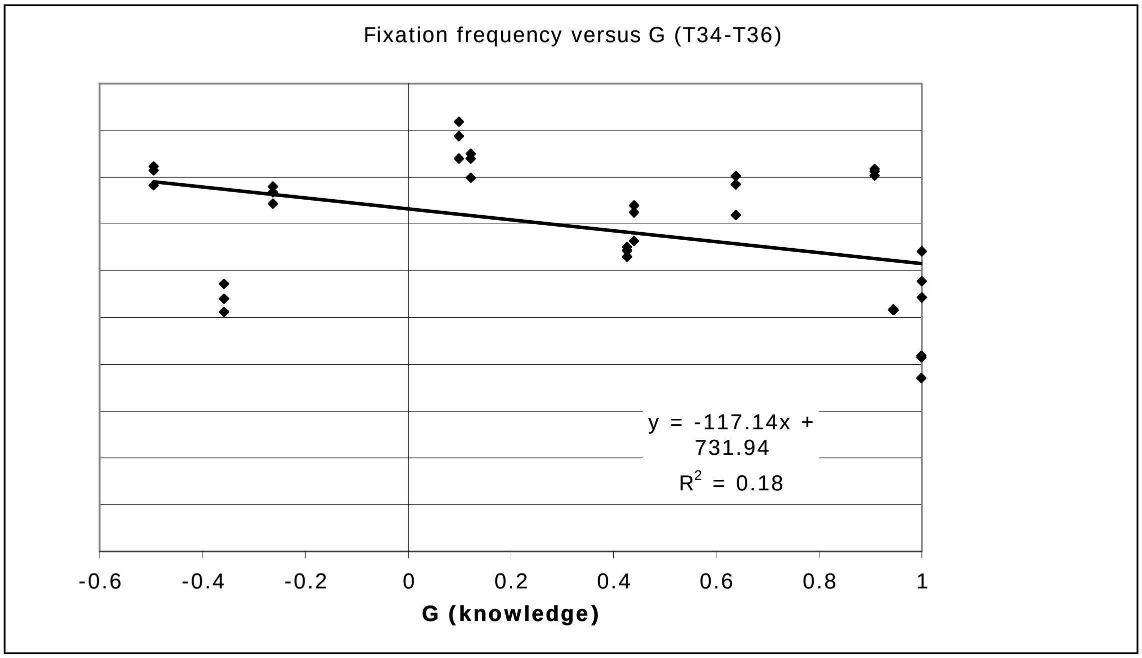


Figure 30. Fixation frequency (count) versus G (knowledge) for T34-T36.

Chapter 6

Summary and Discussion

This study was modeled after Senders (1964) four-dial monitoring task, which suggested that signal bandwidth (BW) drives visual sampling. Also, informed by SEEV model of visual attention allocation (Wickens et al., 2003), value was as an additional sampling cue. Ecological validity, measured as the correlation between the proximal BW cue and the distal task criterion (alarm rate), was added as a measure of the reliability between the interface (BW and value cues) and the environment. One of the primary goals of the thesis was to examine how these proximal cues (BW and value), mediated by ecological validity, affected scanning strategies and performance. More simply, the question we asked was: Do people adapt their scanning strategies based solely to the BW (and value) cue, only to the task criterion (alarm rate x value), or is scanning based on some other combination of these factors? A second goal of this study was to potentially inform and evaluate the SEEV model of visual attention allocation based on the expectancy (BW) and value components of the model.

6.1 Does EV Affect Visual Attention Allocation?

Ecological validity clearly affected visual attention allocation in several expected and unexpected ways. This section will discuss how the EV manipulation influenced learning, which was a sub-goal of the thesis, as well as affected final adaptation to the task.

First, the results summarized in Figure 31 suggest that participants across all EV conditions were able to adapt to the task in terms of performance (total score). By the final day of the study all participants had approximately equivalent performance scores. Adaptation is shown in Figure 31 as an improvement in performance over the course of the study. This improvement suggests that participants in the EV<1 conditions were relying, at least partially, on a cue other than BW. In the two EV<1 conditions (EV=0.75 and EV=0.25), BW was an imperfect predictor of the alarm rate, therefore participants could not reliably use the proximal cue to predict the distal alarm criterion.

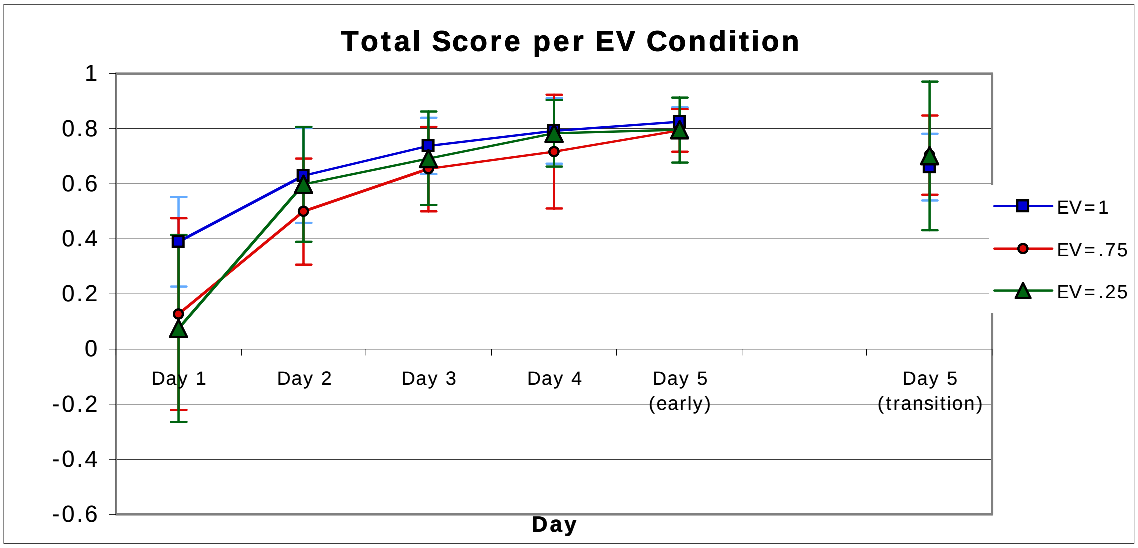


Figure 31. Total score by EV condition (mean per day and participant). The total score is a function of the total points received divided by the total points possible. The error bars indicate one standard deviation. Day 5 (early) indicates the pre-transfer condition and Day 5 (transition) indicates the post-transition results.

However, the degree of EV (i.e. how reliable the BW cue was in predicting alarm rate) affected both sampling strategies and learning in several interesting ways. Not surprisingly, when BW was perfectly correlated with alarm rate (as in Senders 1964) participants had the easiest time learning the task and their scanning strategies were most optimally adapted to the environment by the end of the study. Since there are such large differences between EV conditions early, but not late, this suggests that participants in the EV=1 condition had less to learn over the course of the study than the EV<1 conditions. Since the salient cues (BW and value) were predictive of the task criterion (detecting alarms) in the EV=1 group, participants could perform well by adapting to these proximal cues. It might be said that BW is a salient cue that, at least initially, automatically drives attention.

If BW is indeed a salient cue driving attention, it suggests that participants in the EV=1 condition would be expected to perform better than those in the EV<1 (EV=0.75 and EV=0.25) conditions from the beginning, because the salient cue (BW) was a better predictor of the task criterion than in the EV<1 conditions. Based on this assumption, the EV=0.75 group would also be expected to initially perform better than the EV=0.25 group. However, one could also argue that since BW was only minimally correlated with alarm rate in the EV=0.25, by the later stages of the study, participants may have easily recognized that the cue could not be reliably used to detect alarms (the task criterion) and learned to ignore it. The operators in this condition could then focus on learning the cues that allowed them to successfully adapt to the environment. Since the EV=0.75 group was presented with a BW cue that was more highly correlated with alarm rate, it appears to have required more time for this group to recognize and adapt to the fact that the most salient cue was not a reliable predictor of the task criterion.

Though there was little difference in performance across EV conditions, several additional analyses provided evidence that, in fact, the two EV<1 conditions appeared to be quite different with respect to the manner in which visual scanning was adapted to the task structure. Interestingly, participants in the least predictive environment (EV=0.25) appeared to learn faster and were more adapted to their environment, at least in terms of adaptation of visual attention allocation strategies, by the end of the study than participants in a somewhat reliable environment (EV=0.75). Figures 32, 33, and 34 all provide evidence that participants in the EV=0.25 condition were more optimally adapted to the environment than the EV=0.75 group with respect to visual attention allocation.

Figure 32 illustrates the apparent difference between the EV conditions by comparing the average number of misses as a function of the additive value of the BW and value cues per gauge over the final three pre-transfer trials (T34-T36). It was expected that reducing the number of misses on gauges with higher values (BW + value) implied better adaptation to the task. Based on this assumption, clearly the EV=1 group was better adapted to recognizing which gauges were more valuable (or costly), while the EV=0.25 group was somewhat adept at missing alarms on the less-valuable gauges. The EV=0.75 group did not appear to be well adapted to the BW and value combination at this final stage of the study.

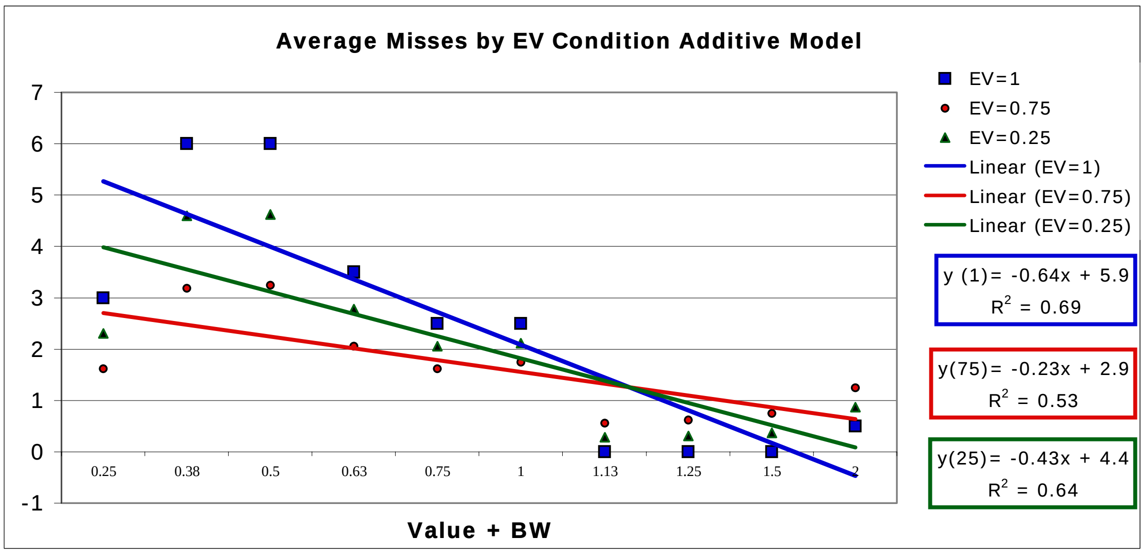


Figure 32. Average misses by EV (additive combination of BW and value) for Trials 34-36.

Further evidence that the EV=0.75 group was not as adapted to the task as the EV=0.25 condition was provided in the comparison of the model predicted (optimal) versus observed dwell durations per gauge over trials 34 through 36, and is summarized in Figure 33. In the figures, a slope of 1 represented optimal adaptation. From Figure 33, it is quite clear that the EV=0.75 group was not at all optimally adapted, in terms of visual scanning strategies, to the task. Their scanning pattern appeared to be completely random, with a slope and an R^2 value of zero. This means that the fixation frequencies were all around 25% across participants in this condition.

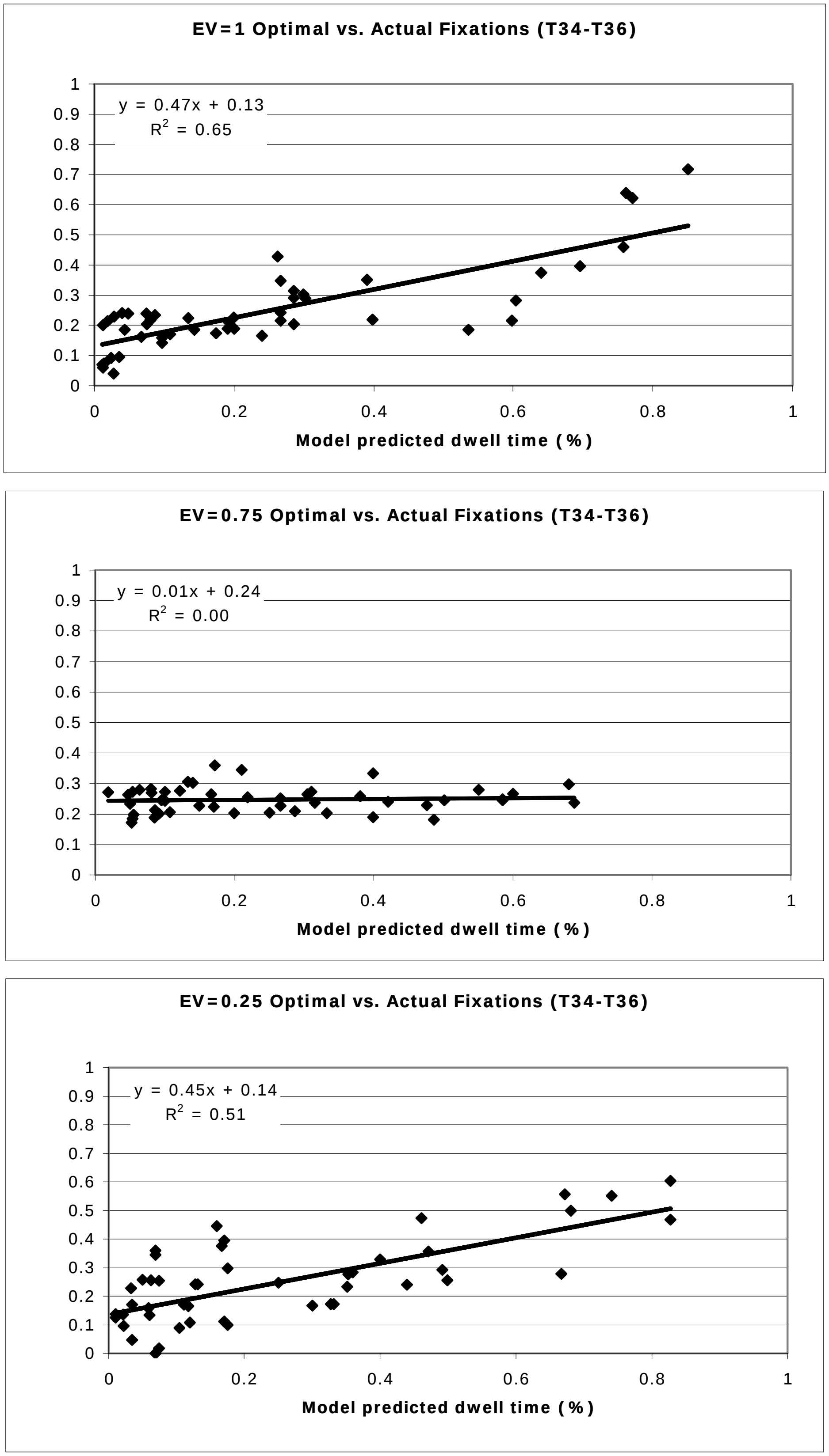


Figure 33. Summary of model versus observed dwell times per EV condition.

Finally, Figure 34 presents additional support that the EV=0.75 group was clearly negatively affected by the partially predictable proximal environment. The Lens Modeling revealed that participants in the EV=0.25 group and the EV=1 group were much more optimally adapted to their respective environments (in terms of scanning not performance) than the EV=0.75 group. At the final stages of the study, it appears that participants had no linear or unmodeled knowledge about the task, and as shown above with the correlation between optimal and observed dwell durations, achievement was also close to zero.

Additionally, the cue weightings shown in Figure 34 suggest that participants in the EV=0.75 group were not appropriately weighting the value cue. It could be argued that value is a less salient cue than BW. Since participants in the EV=0.75 condition were still dealing with the reliability of the salient BW cue, it is quite possible that they could not yet adapt to properly weighting the value cue. It appears that participants in the EV=1 and EV=0.25 conditions were, at least partially, weighting the value cue.

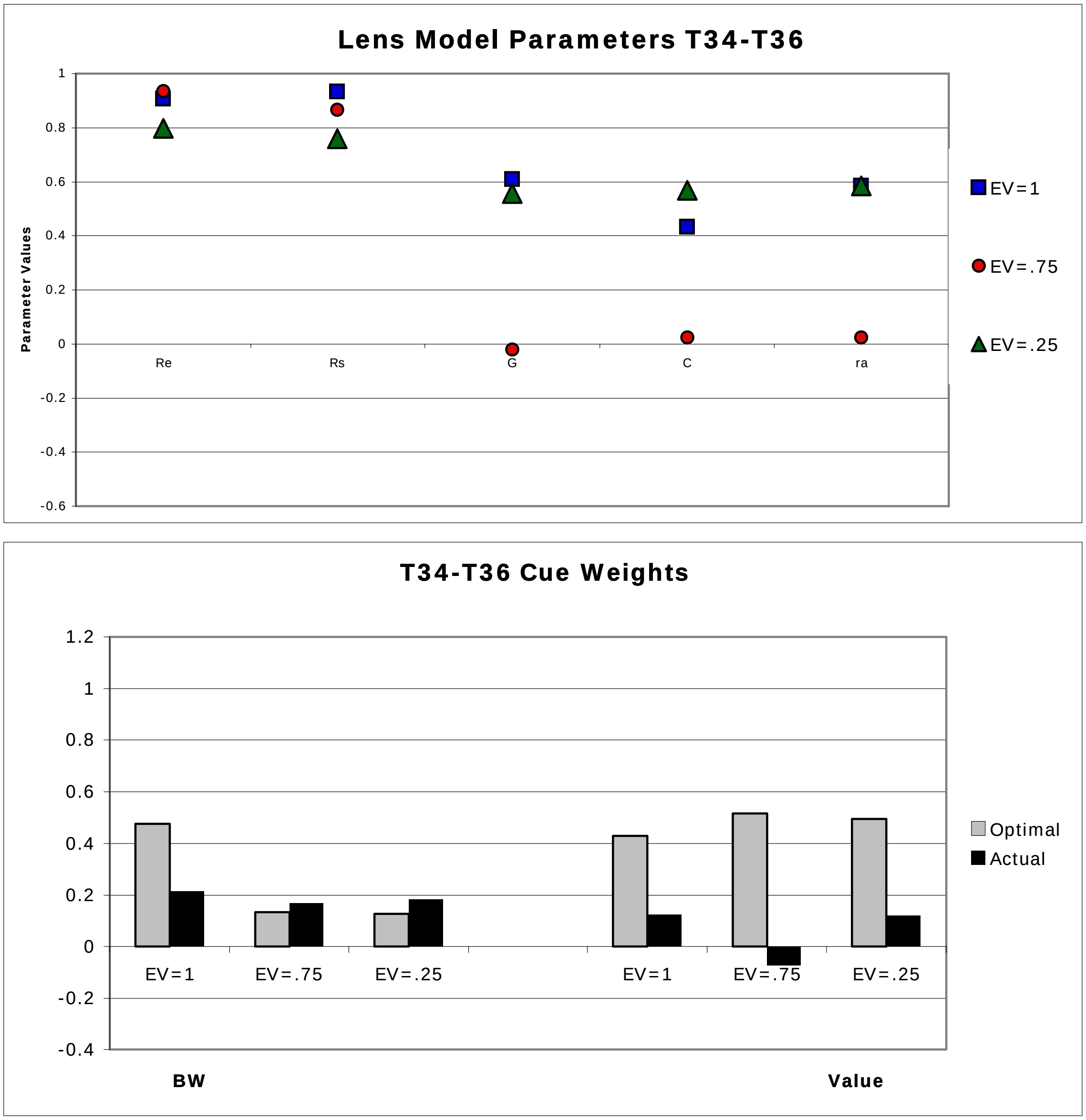


Figure 34. Lens Model parameters and cue weights for T34-T36 averaged per EV condition.

Note that the averages are based on participants who could be reliably modeled for each condition. All participants produced significant models for the final set of trials.

6.2 Implications for the SEEV Model

The results from this study (and discussed in Byrne and Kirlik, 2004) found that an additive rather than a multiplicative formulation of the SEEV model provided a better fit for the data. Although the analysis by Byrne and Kirlik (2004) regarding the formulation of the expected value portion of the SEEV model as additive contrasts with the current model predictions that combine expectancy and value multiplicatively, the findings are not entirely unexpected when examined from a judgment framework. There is converging evidence that linear models accurately capture people's judgment polices better than those involving multiplicative functions (Arkes and Hammond, 1986). Arkes and Hammond (1986) suggest that tasks that involve curviliniear forms are more difficult to learn and more difficult for others to understand.

Although the current study is not a judgment problem in the traditional sense, it should not be difficult to see the link between combining cues to make a judgment and combining cues (BW, Value) to direct visual attention. Information acquisition, inferred through eye movements, can be either internally or externally driven depending on the circumstances, which should dictate whether the combination of cues are combined linearly (BW + Value) or if the situation is best modeled with a multiplicative formulation (BW x Value). Since tasks that involve curvilinear forms are difficult to learn, it would be expected that combining cues in this way would require an internalized knowledge-driven process. An example of an internally knowledge driven process would be our earlier example (in the introduction) of where the person was searching for news on the internet. However, additive (linear) combinations of cue weightings are more likely to be observed in real-time externally driven situations, as was the case in the present study.

If the SEEV model is indeed best formulated using an additive rather than a multiplicative formulation, the next logical question then becomes whether this result is generalizable to modeling visual attention allocation. Senders (1983), in a similar paradigm, raised a similar point:

> Since subjects received as many as thirty hours of exposure to the array of instruments during the experiments, there was what would in normal circumstances be considered adequate time to learn the signal statistics. However, since the real life tasks to which we wish to extrapolate our results may involve thousands of hours of experience before competence is recognized, our results may be valid only in the laboratory environment. …In the four-dial experiments the relative frequencies of fixation on the various dials are roughly in accord with the model after about three hours of exposure. However, although the sampling was at the 'correct' frequencies, it was not effective for the detection of deviant readings on the instruments as indicated by a relatively low detection rate. Thus, something was lacking in the performance at three hours. It is probably the case that something was still lacking after as many as thirty hours of experience. (pp. 86-87)

This would suggest, at least in our study, that the additive formulation of the SEEV model was not entirely unexpected, because sampling was most likely not yet a knowledge-driven process. It does not, however, allow us to generalize this finding to all situations in which the SEEV model could be applied. However, for pilots with many hours of experience, where sampling might be more knowledge-driven, the expected value formulation may indeed be the more accurate formulation.

Therefore, we propose that future models of visual attention allocation should consider both additive and multiplicative combinations of BW (or expectancy) and value. In addition, as explained in the previous section another factor that deserves consideration when modeling visual attention allocation is the informativeness or reliability of the display (i.e. ecological validity of the proximal information sources).

6.3 Implications for Lens Modeling

In addition to addressing the original study goals, this thesis has also presented a new application of the Lens Model into the visual attention domain. Although this was not one of the original study goals, this modeling technique has, to the best of our knowledge, never been applied in modeling visual attention. Therefore, it is relevant to discuss the potential of Lens Modeling to be used as a tool for studying visual attention allocation and for its application beyond judgment theory.

The Lens Modeling technique proved to be a useful tool in informing the study goals and several central findings that emerged from the analysis. First, the modeling provided additional evidence that EV influenced visual scanning patterns. Second, there appeared to be a relation between performance (as measured by total score) and the achievement (r_a) parameter in the Lens Model (with a few caveats). Sometimes there appeared to be a discrepancy between performance and achievement in terms of adaptivity of visual scanning strategies to the environment. It was found that participants who had a high fixation frequency did not need to be as optimal in their scanning strategies to achieve high performance in the task. Therefore, in order to predict performance using the Lens Model parameters, one also needs to also consider the fixation frequency. A better predictor of individual performance, by the end of the study, appeared to be the correlation between the weighting of the BW cue in relation to the optimal environmental weighting. Those participants who performed the best (in terms of total score) in each EV condition were those who most optimally weighted the BW cue.

The Lens Model analysis did not appear to be successful for investigating the effects of EV on learning both within and across trials. In the early trials, where most performance improvement was taking place, the Lens Model parameters did not appear to change very much. It is possible that either this modeling technique is not appropriate for investigating learning or perhaps a more detailed model would provide a more adequate analysis (see e.g. Skill Score decomposition; Cooksey, 1996).

Chapter 7

Conclusions

The primary motivation for this study stemmed from work modeling visual attention allocation for evaluation of Synthetic Vision System displays when the interface provided unreliable information to the pilot. Models of visual attention have always assumed the interface (the proximal environment) was perfectly predictive and provided completely reliable information about the distal environment. The main goal of this study was to try to understand how people adapt their scanning strategies when presented with unreliable information.

The performance results and modeling analysis revealed that highly reliable, but slightly imperfect automation appears to be worse, in terms of scanning, than interfaces that provide relatively poor predictability. Those presented with a highly predictive but imperfect environment were found to be far less adapted to the task than those participants who were presented with cues that were not reliable with respect to the task criterion. Notably, by the end of the study, performance, in terms of meeting the task goals, was the same across ecological validity conditions. On the surface, this seems to suggest that ecological validity is not a factor in monitoring performance. However, the lack of adaptation in the highly reliable but imperfect condition manifested itself in other, less readily apparent, differences in scanning behavior across conditions where the display was either perfectly reliable or highly imperfect.

The results from this study speak to the potential problems associated with introducing, *slightly* unreliable SVS displays into the airplane cockpit. Also, this study lends additional support to a growing body of literature speaking to the (sometimes unintended) effects of imperfect automation on human behavior (see Parasuraman and Riley, 1997 for a review). Achieving the positive benefits associated with new forms of automation without suffering the costs associated with their potential imperfections, however slight, promises to remain a challenge for some time to come.

Bibliography

Arkes, H. R. and Hammond, K.R. (1986). Judgment and decision making: An interdisciplinary reader. Cambridge, Cambridge University Press.

Bellenkes, A., Wickens, C.D., and Kramer, A.F. (1997). "Visual Scanning and Pilot Expertise." Aviation Space and Environmental Medicine **48**(7): 569-579.

Bisantz, A. M., Kirlik, A., Gay, P., Phipps, D.A., Walker, N., and Fisk, A.D. (2000). "Modeling and analysis of a dynamic judgment task using a lens model approach." IEEE Transactions on Systems Man and Cybernetics Part a-Systems and Humans **30**(6): 605-616.

Bohnen, H. G. M. and Leermakers, M. A. M. (1991). "Sampling behavior in a four instrument monitoring task." IEEE Transactions on Systems Man and Cybernetics **SMC-21**: 893-897.

Bohnen, H. G. M., Leermakers, M. A. M., and Venemans, P.J. (1996). "Sampling behavior in a four instrument monitoring task: effects of signal bandwidth and number of events per signal." IEEE Transactions on Systems Man and Cybernetics, Part A **26**(4): 413-422.

Brunswik, E. (1956). Perception and the representative design of the psychological experiments. Berkeley, CA, University of California Press.

Byrne, M. D. and Kirlik, A. (2004). Integrated Modeling of Cognition and the Information Environment: A Multilevel Investigation (Process and Product Modeling) of Attention Allocation to Visual Displays. (Technical Report AHFD-04-14/NASA-04-4) Savoy, IL: University of Illinois, Institute of Aviation.

Carbonell, J. R. (1966). "Queueing Model of Many-Instrument Visual Sampling." Transactions on Human Factors in Electronics **HFE-7**(4): 157-164.

Cooksey, R. W. (1996). Judgment Analysis: Theory, Methods, and Applications. New York, Academic.

Craik, K. J. W. (1947). "Theory of the Human Operator in Control Systems." British Journal of Psychology **38**: 56-61.

Fitts, P. M., Jones, R. E., and Milton, J.L. (1950). "Eye movements of aircraft pilots during instrument landing approaches." Aeronautical Engineering Review **9**: 1-5.

Glaab, L. J. and Takallu, M. A. (2002). Preliminary Effect of Synthetic Vision Systems Displays to Reduce Low-Visibility Loss of Control and Controlled Flight Into Terrain Accidents. Hampton, VA, NASA Langley Research Center.

Hammond, K. R. and Stewart, T. R. Eds. (2001). The Essential Brunswik: Beginnings, Explications, and Applications. New York, Oxford University Press.

Holland, J. H. (1992). Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT Press.

Hopkin, V. D. (1982). Human factors in air-traffic control. Brussels, NATO.

Kvalseth, T. A. (1977 (a)). The effect of cost on the sampling behavior of human instrument monitors. Monitoring behavior and supervisory control. T. B. Sheridan and G. Johannsen. New York, Plenum.

Moray, N. (1981). The role of attention in the detection of errors and the diagnosis of failures in man-machine systems. Human detection and diagnosis of system failures. J. Rasmussen and W. B. Rouse. New York, Plenum.

Moray, N. (1986). Monitoring Behavior and Supervisory Control. Handbook of perception and human performance. K. Boff, I. Kaufmann and J. Beatty. New York, Wiley. **H**.

Moray, N. (1990). "Designing for transportation safety in the light of perception, attention, and mental models." Ergonomics **33**(10-11): 1201-1213.

Mumaw, R. J., Sarter, N. B., and Wickens, C.D. (2001). Analysis of pilots' monitoring and performance on an automated flight deck. 11th International Symposium on Aviation Psychology, Columbus, OH.

Parasuraman, R., and Riley, V. (1997). "Humans and automation: Use, misuse, disuse, abuse." Human Factors **39**(2), 230-253.

Parasuraman, R., Sheridan, T. B., and Wickens, C. D. (2000). A model for types and levels of human interaction with automation. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 30(3), 286-297.

Raby, M. and Wickens, C. D. (1994). "Strategic workload management and decision biases in aviation." International Journal of Aviation Psychology **4**(3): 211-240.

Senders, J. W. (1964). "The Human Operator as a Monitor and Controller of Multidegree of Freedom Systems." IEEE Transactions on Human Factors in Electronics.

Senders, J. W. (1983). Visual Scanning Processes. Netherlands, University of Tilburg Press.

Senders, J. W., Kristofferson, A.B., Levison, W.H., Dietrich, C.W., and Ward, J.L. (1967). “The attentional demand of automobile driving.” Highway Research Record, **195**: 14-33

Sheridan, T. B. (1970). "On how often the supervisor should sample." IEEE Transactions on Systems Man and Cybernetics **SSC-6**: 140-145.

Stewart, T. R. and Lusk, C. M. (1994). Seven components of judgmental forecasting skill: Implications for research and the improvement of forecasts. Journal of Forecasting, **13**, 575-599. (Reprinted in Connolly, T., Arkes, H. R., and Hammond K. R., Eds., (2000) Judgment and Decision Making: An Interdisciplinary Reader, Second Edition. New York: Cambridge University Press.)

Strauss, R. and Kirlik, A. (in press). "A systems perspective on situation awareness II: Experimental evaluation of a modeling and measurement technique." International Journal of Industrial Ergonomics.

Wickens, C. D., Goh, J., Helleberg, J, Horrey, W.J., and Talleur, D.A. (2003). "Attentional models of multitask pilot performance using advanced display technology." Human Factors **45**(3): 360-380.

Wickens, C. D., Helleberg, J., Goh, J., Xu, X., and Horrey, W.J. (2001). Pilot Task Management: Testing an Attentional Expected Value Model of Visual Scanning. Savoy, University of Illinois Institute of Aviation.

Wickens, C. D., Helleberg, J., Xu, X. (2002). "Pilot maneuver choice and workload in free flight." Human Factors **44**: 171-188.

Wickens, C. D. and Hollands, J.G. (2000). Engineering Psychology and Human Performance. Upper Saddle River, NJ, Prentice Hall.

Yantis, S. (1993). "Stimulus-driven attentional capture and attentional control settings." Journal of Experimental Psychology: Human Perception and Performance **19**: 676-681.

Appendix A

Participant Instructions

All participants were read the following instructions, followed by an exchange it which the experimenter answered any (permissible) questions that the participants had about the task.

For this study we are trying to understand the factors that affect where you decide to direct your visual attention. We are hoping to use this information to guide display design in airplane cockpits, so it is very important that you try and do your best throughout the study. We will be giving the top performer in each group (of five people) a $50 bonus. I will tell you more about that when I explain what you will be doing.

It is important that you attend your session each day. If you cannot make your scheduled session please let me know as soon as possible, and we can try and re-schedule another time in the day. For this study we are asking you to complete 8 five-minute trials per day for the entire week. You will have a short break following each session and a longer break in the middle. We will make a video recording of each session to look at your eye movements.

To obtain accurate eye-movement measures, it is necessary that you are a predetermined distance from the computer monitor. We have set up the computer monitor so that it is the correct distance away from your eyes when you are lined up with this string. … Please try and keep this distance from the monitor during the entire set of trials. We have also set it up that the correct distance is when your eyes are lined up with the edge of the desk. If you are not the correct distance from the monitor during one of your trials, we will let you know.

At the beginning of each trial you will see a box that asks you to enter your ID. You don't need to put anything in this box. Pressing start will begin your session. Please make sure an experimenter is in the room and tells you to begin before you press start.

As you can see there are four gauges on the screen. Each gauge has a pointer that is initially pointing upwards. Once you press start, all of these dials will begin to move back and forth. There is a safe range for all the values as indicated by the light gray areas, and an alarm range as indicated by the dark gray areas. Your goal is to detect when each of the gauges goes into the alarm range by pressing a button on the keyboard corresponding to the gauge that is currently out of bounds. You have one second to press the button once the gauge has gone out of bounds. If the upper left gauge is out of range, press the "a" key. The lower left gauge corresponds to the *"z" key. If the upper right is out of bounds press the " ' " key. Lower right is the "/" key. You only need to press the button once each time the pointer goes out of range. If you press it more than once each time it goes out of bounds, it will be considered a false alarm (which we will go over later). So, your overall goal of this experiment is to monitor all four gauges continuously and detect when any one of them goes out of bounds.*

Also, in the alarm range of each gauge, there are bars of different lengths. These bars are used to indicate the number of points each gauge is worth. The longest bar is worth 8 points, the bigger medium-sized bar is worth 4 points, the smaller bar is worth 2, and the smallest bar is worth 1 point. Each time you correctly detect that a gauge has gone out of bounds you will receive the number of points indicated by the length of the bar. There are two ways to loose points. If you miss that an alarm has gone out of range you will loose the number of points based on the value of the gauge. You will also loose the same number of points, based on the value of the gauge, when you press the key indicating that the gauge has gone out of bounds when it is still in the safe zone (light gray). If you press a key more than once when the alarm is out of bounds, you will not receive more points. This is considered to be a false alarm, so you will loose points.

Your overall score will be the positive points you earn for each time you correctly detect that a gauge has gone out of bounds minus the points you loose for both misses and false alarms. Since there are many ways to loose points, it is possible (or even likely) that you will have negative points – especially at the beginning.

Your total score will be given to you at the end of the each trial.

To calculate your score for the bonus at the end, we will take the average of your top five scores. over the whole week. The top performer in each group (1 in 5 chance) will receive an extra $50.

Appendix B

Task Program

```
// This program is meant to simulate Sender's experiment according to:
// John W. Senders, The Human Operator as a Monitor and Controller
// of Multidegree of Freedom Systems
//
// (C) 2003,2004 Alex Kosorukoff <kosoruko@uiuc.edu>
// V0.2 UDP controller added
// $Revision: 1.97 $

import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.net.*;
import java.text.*;
import java.util.*; // for Date

public class SendersTest extends Frame {
  static boolean fullscreen = false;  // determines a window size
  static boolean logging = true;      // write log files
  static boolean verbose = true;      // print messages showing progress
  static int test_duration = 1;       // in minutes
  double tcorr = 0.25, trate = 0.25;  // target parameters of experiment
  String ppfile = "Population.ser";   // file to store parameter population
  static int iterations = 25;         // number of ES fitting iterations
  static int timer_interval = 1000/20;
  static int degree = 4;
  static String bcommand = null, ecommand = null;

  class Control extends UDPTransceiver {
      public Control(int port, String hostname, int hport)
        throws IOException {
        super(port,hostname,hport);
```

```
    }
    public void Notify(String s) {
      // modifies SendersTest class parameters
      System.out.println(s); // debug stub, prints a command
      if(s.substring(0,1).equals("I")) { // initialize a session
            host = dpr.getAddress(); hport = dpr.getPort();
            name = s.substring(2); // need to change it later

            System.out.println("Initializing session for "+name);
            System.out.println("Sending acq to "+host+" port "+hport);
            //tm.t = 0;
            //tm.running = true;
            if(st!=null) {
              suppress_id = true;
              st.hide();
              st.dispose();
            } else {
              id.tf.setText(name);
              id.hide();
            }
            send("K");
      } else if(s.substring(0,1).equals("P")) { // operator pressed a key
            send("K");
            warning("UDP");
      } else if(s.substring(0,1).equals("G")) { // get thetas in rad
            String ds = "";
            for(int i=0; i<g.length; i++)
              ds += g[i].theta + " ";
            send(ds);
      } else if(s.substring(0,1).equals("g")) { // get thetas in deg
            String ds = "";
            for(int i=0; i<g.length; i++)
              ds += (int)(g[i].theta*180/Math.PI) + " ";
            send(ds);
      } else if(s.substring(0,1).equals("E")) { // end session
            tm.running = false;
            send("K");
            System.out.println("Finalizing session");
      }
    }
}
Timer tm;
Control ctr;
static InitDlg id;
static Dialog  st;
static boolean suppress_id = false;
```

```
    int port;
    static String name;
    Gauge g[];

    public SendersTest() {
        this(new Dimension(480,480));
    }
    public SendersTest(Dimension dim) {
        setSize(dim);
        setTitle("Sender's Test");
        addWindowListener( new WindowAdapter() {
                public void windowClosing(WindowEvent e) { System.exit(0); }
          });
        addKeyListener( new KeyAdapter() {
                public void keyPressed(KeyEvent e) {
                  warning(e);
                }
          });


        Class gc = null;
        String gname = "RoundGauge";
        //String gname = "RectGauge";
        try { gc=Class.forName(gname); }
        catch(ClassNotFoundException e) {
          System.out.println("No gauge class found: "+e);
          System.exit(1);
        }

        //Gauge
        Gauge.degree = degree;
        g = new Gauge[4];
        for(int i=0; i<4; i++) {
          g[i] = new RoundGauge();
          g[i].id = i;      // assign ids
        }
//      try {
//        for(int i=0; i<4; i++)
//              g[i] = (Gauge) gc.newInstance();
//      } catch(Exception e) {
//        System.err.println("Can't instantiate class: "+e);
//        System.exit(1);
//      }

        g[0].setparam(200,1,0.5);
        g[1].setparam(200,2,1.0);
```

```
        g[2].setparam(200,4,2.0);
        g[3].setparam(200,8,4.0);
        //for(int i=0; i<g.length; i++) g[i].maxlatency = 20*5; //for testing


        // Main window layout

    setLayout(new GridLayout(2,2,50,50));
        for(int i=0; i<g.length; i++) add(g[i]);
//      pack();

//      GridBagLayout gbl = new GridBagLayout();
//      setLayout(gbl);

//      for(int i=0; i<g.length; i++) {
//         GridBagConstraints gbc = new GridBagConstraints();
//         gbc.gridx = i%2; gbc.gridy = i/2;
//         gbc.gridwidth = 1; gbc.gridheight = 1;
//         gbc.fill = GridBagConstraints.NONE;
//         //gbc.ipadx = 10; gbc.ipady = 10;
//         gbc.insets = new Insets(3,3,3,3);
//         //gbc.anchor = GridBagConstraints.CENTER; // default
//         gbc.weightx = 10; gbc.weighty = 10;
//         gbl.setConstraints(g[i],gbc);
//         add(g[i]);
//      }

        //pack();
        // initialize initial dialog
        id = new InitDlg(this);

        // initialize UDP interface
        for(port = 49000; port<49002; port++) {
           try {
                 ctr = new Control(port,"localhost",port==49000?49001:49000);
                 if(verbose)
                    System.out.println("Listening "+ctr.port+" sending to "+ctr.hport);
                 break;
           } catch(IOException e) {
                 System.err.println(e);
           }
        }
        if(ctr!=null) ctr.start();
        else {
           System.err.println("Can't establish UDP connection");
           System.exit(1);
```

```
        }


        addMouseListener( new MouseAdapter() {
                public void mousePressed(MouseEvent e) {
                    warning(e.getSource()); }
            });

        for(int i=0; i<g.length; i++) {
            g[i].addMouseListener( new MouseAdapter() {
                    public void mousePressed(MouseEvent e) {
                        warning(e.getSource()); }
                });
            g[i].addKeyListener( new KeyAdapter() {
                    public void keyPressed(KeyEvent e) {
                        //warning(e.getSource());
                        warning(e);
                    }
                });
        }

        tm = new Timer(g,timer_interval);
        tm.start();
        tm.running = false;
    }
    public double ape() { // alarm probability estimate
        double ape = 0;
        for(int i=0; i<g.length; i++) ape += g[i].ape();
        return ape;
    }
    public void info() {
        info(System.out);
    }
    public void info(PrintStream ps) {
        NumberFormat fp = new DecimalFormat(" 0.00;-0.00");
        ps.println("# Target: tcorr = "+fp.format(tcorr)+
                   " trate = "+fp.format(trate));
        ps.print("#\n#");
        for(int s=0; s<g[0].degree; s++) {
            ps.print("   Am"+s);
            ps.print("   Ph"+s);
        }
        ps.println("");

        for(int i=0; i<g.length; i++) {
            ps.print("# ");
```

```
            for(int s=0; s<g[i].degree; s++) {
                ps.print(fp.format(g[i].amp[s])+" ");
                ps.print(fp.format(g[i].phase[s])+" ");
            }
            ps.println("\t["+i+"]");
        }
        ps.println("#");

        for(int i=0; i<g.length; i++)
            ps.println("# Gauge "+i+
                        " ape = "+fp.format(g[i].ape())+
                        " bnd = "+fp.format(g[i].bandwidth));
        double corr = correlation();
        double rate = ape();
        double dist = // best distance of known solution
            Math.sqrt((tcorr-corr)*(tcorr-corr)+(trate-rate)*(trate-rate));
        for(int i=0; i<4; i++) // penalty
            if(g[i].ape()==0 || g[i].mmgap<0.1*g[i].limit) dist++;
        //System.out.println("Minmaxgaps: ");
        //for(int i=0; i<4; i++)
        //    System.out.print(fp.format(g[i].mmgap/g[i].limit)+"\t");
        //System.out.println("");

        ps.println("#\n# Corr = "+fp.format(corr)+
                    "; rate = "+fp.format(rate)+
                    "; d = "+fp.format(dist));
    }

    public double fit() {
        // Using evolutionary strategy ES(1,1,0) to get as close as we can
      // to the target (tcorr,trate) in n generations

        // Retrieve or create the population parameter file
        // it is good to move it into constructor later
        // String ppfile = "Population.ser";
        int[] bwperm = new int[4];          // temporary hack
        for(int i=0; i<4; i++) {
            int bwi = (int) Math.round(1+Math.log(g[i].bandwidth)/Math.log(2));
            bwperm[bwi] = i;                // inverse bw mapping
        }

        Population pp;
        if (new File(ppfile).exists()) {
            try {
                FileInputStream fis = new
                    FileInputStream(ppfile);
```

```
            ObjectInputStream ois = new
               ObjectInputStream(fis);
            pp = (Population)ois.readObject();
            fis.close();
            //System.out.println("Pop: "+pp);
      } catch(Exception e) {
            System.err.println(e);
            pp = new Population(); // this might lead to loosing pop
      }
   } else {
      System.out.println("No solution file found ["+ppfile+"]");
   pp = new Population();
   }

   boolean details = verbose && false;    // print while fitting
   double qdist = 1000;
   NumberFormat fp = new DecimalFormat(" 0.00;-0.00");

   // Replace parameters by saved solutin (if any)
   try {
      Signal[] sig =
            (Signal[]) pp.pop.get(new Goal(test_duration,trate,tcorr));

      System.out.println("Saved solution in file ["+ppfile+"]");
      for(int i=0; i<4; i++) System.out.println(sig[i]);

      for(int i=0; i<4; i++)
            g[bwperm[i]].setparam(sig[i]);

      double corr = correlation();
      double rate = ape();
      double dist = // best distance of known solution
            Math.sqrt((tcorr-corr)*(tcorr-corr)+(trate-rate)*(trate-rate));
      for(int i=0; i<4; i++) // penalty
            if(g[i].ape()==0 || g[i].mmgap<0.1*g[i].limit) dist++;
      if(false) {
            System.out.println("Minmaxgaps: ");
            for(int i=0; i<4; i++)
               System.out.print(fp.format(g[i].mmgap/g[i].limit)+"\t");
            System.out.println("");
      }
      if(verbose)
            System.out.println("Saved solution ["+ppfile+"]:\n"+
                              "Corr = "+fp.format(corr)+
                              "; rate = "+fp.format(rate)+
                              "; d = "+fp.format(dist));
```

```
    qdist = dist;
} catch(NullPointerException e) {
    System.out.println("No saved solutions");
};

if(verbose)
    System.out.println("\nFitting target ("+tcorr+", "+trate+")");

//Gauge qopt[] = (Gauge[]) g.clone(); // doesn't work: persist str.
//Gauge qopt[] = g;    // quasi-optimal parameter set
Gauge qopt[] = new Gauge[g.length];
for(int i=0; i<g.length; i++) qopt[i] = (Gauge) g[i].clone();

//iterations = 2; //for debugging
for(int fi=0; fi<iterations; fi++) { // fiddle with parameters
    // Randomization
    if(details) System.out.println("Functional parameters:");
    for(int i=0; i<4; i++) {
        g[i]._ape = -1; // invalidate old ape
        if(details) System.out.print("["+i+"] ");
        for(int s=0; s<g[i].degree; s++) {
            //g[i].amp[s] = (1.2+0.5*(1-2*Math.random()))/degree;
            g[i].amp[s] = (1.5+(1-2*Math.random()))/degree;
            //g[i].phase[s] = Math.random()*2*Math.PI;
            g[i].phase[s] = Math.random()<0.5? 0 : Math.PI;
            g[i].phase[s] = (2*Math.random()-1)*Math.PI/8;
            if(details) System.out.print(fp.format(g[i].amp[s])+" "+
                                          fp.format(g[i].phase[s])+" ");
        }
        if(details) System.out.println("\t["+i+"]");
    }
    if(details) System.out.println("");

    double corr = correlation();
    double rate = ape();
    double dist =
        Math.sqrt((tcorr-corr)*(tcorr-corr)+(trate-rate)*(trate-rate));
    for(int i=0; i<4; i++) // penalty
        if(g[i].ape()==0 || g[i].mmgap<0.1*g[i].limit) dist++;
    if(false) {
        System.out.println("Minmaxgaps: ");
        for(int i=0; i<4; i++)
            System.out.print(fp.format(g[i].mmgap/g[i].limit)+"\t");
        System.out.println("");
    }
    if(verbose)
```

```
            System.out.print("Corr = "+fp.format(corr)+
                            "; rate = "+fp.format(rate)+
                            "; d = "+fp.format(dist));

        if(dist<qdist) {
            if(verbose) System.out.println(" !");
            //qopt = (Gauge[]) g.clone(); // this doesn't work
            for(int i=0; i<g.length; i++) qopt[i] = (Gauge) g[i].clone();
            qdist = dist;
        } else
            if(verbose) System.out.println("");
    }

    //g = qopt; // retrieve the best match
    for(int i=0; i<g.length; i++) {
        g[i].amp = qopt[i].amp;     // retrieve fit amp
        g[i].phase = qopt[i].phase; // retrieve fit phase
        g[i].freq = qopt[i].freq;   // retrieve the set of frequencies
        g[i]._ape = -1;             // invalidate old ape
    }

    if(verbose)
        System.out.println("\nError = "+fp.format(qdist)+
                        " after "+iterations+" iterations\n");
    // Update saved solutions
    Signal[] sig = new Signal[4];
    for(int i=0; i<4; i++)
        sig[i] = new Signal(g[bwperm[i]]); // bwperm

    //for(int i=0; i<4; i++) System.out.println(sig[i]);

    pp.pop.put(new Goal(test_duration,trate,tcorr), sig);

    // Saving solutions found
    System.out.println("Saving "+ppfile);
    try {
        FileOutputStream fos = new FileOutputStream(ppfile);
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(pp);
        oos.close();
    } catch(Exception e) {
        System.err.println(e);
    }

    return qdist;
}
```

```
public double correlation() { // correlation(ape,bandwidth)
        double sum_ape = 0, sum_bnd = 0;
        double ssq_ape = 0, ssq_bnd = 0;
        int test_time_ms = test_duration*60*1000;
        for(int i=0; i<g.length; i++) {
            g[i].ape(test_time_ms,timer_interval,g[i].limit);
            sum_ape += g[i].ape();
            sum_bnd += g[i].bandwidth;
            ssq_ape += g[i].ape()*g[i].ape();
            ssq_bnd += g[i].bandwidth*g[i].bandwidth;
        }
        // calculating correlation(ape,bnd)
        double am = sum_ape/g.length, bm = sum_bnd/g.length;
        double as = Math.sqrt(ssq_ape/g.length-am*am);
        double bs = Math.sqrt(ssq_bnd/g.length-bm*bm);
        double covs = 0, c_den = 0;
        for(int i=0; i<g.length; i++) {
            covs += (g[i].ape() - am)*(g[i].bandwidth - bm);
            c_den += Math.abs((g[i].ape() - am)*(g[i].bandwidth - bm));
        }
        double cov  = covs/g.length;
        double corr = covs/c_den;
        double correlation = cov/(as*bs);
        return correlation;
}
public void warning(Object s) {
        tm.detected++;
        System.out.println("warning issued by "+s);
        if(s instanceof Gauge) {
            //System.out.println(((Gauge)s).id);
            tm.alarm_id = ((Gauge)s).id;
        } else
            tm.alarm_id = -1;
        if(s instanceof KeyEvent) {
            int key = ((KeyEvent)s).getKeyCode();
            switch(key) {
            case 65: tm.alarm_id = 0; break;
            case 222: tm.alarm_id = 1; break;
            case 90: tm.alarm_id = 2; break;
            case 47: tm.alarm_id = 3; break;
            default: tm.alarm_id = -1;
            }
            System.out.println("key "+key+" -> alarm"+tm.alarm_id);
        }
}
public void set_val(String vals) {
```

```
        StringTokenizer vl_tk = new StringTokenizer(vals,",");
        if(vl_tk.countTokens()>0) {
           for(int i=0; vl_tk.hasMoreTokens(); i++)
                 try {
                    g[i].value = Double.parseDouble(vl_tk.nextToken());
                 } catch(NullPointerException e) {};
        }
}
public void set_bws(String bws) {
        StringTokenizer vl_tk = new StringTokenizer(bws,",");
        if(vl_tk.countTokens()>0) {
           for(int i=0; vl_tk.hasMoreTokens(); i++)
                 try {
                    g[i].bandwidth = Double.parseDouble(vl_tk.nextToken());
                 } catch(NullPointerException e) {};
        }
}
public static void wait(int sec) {
        try { Thread.sleep(sec*1000); }
        catch(InterruptedException e) {
           System.out.println("Exception in sleep: "+e);
        }
}
// this function is for Mac only
//public static void paint(Graphics g) {
//g.setColor(getBackground());
//g.fillRect(0,0,getSize().width,getSize().height);
//}
public static void main(String argv[]) {
        SendersTest win = new SendersTest();
        double tcorr_list[] = null;
        String ff_list[] = null;
        String vl_list[] = null;
        String bw_list[] = null;
        try {
           Properties defprops = new Properties();
           defprops.put("fullscreen", win.fullscreen?"true":"false");
           defprops.put("logging", win.logging?"true":"false");
           defprops.put("verbose", win.logging?"true":"false");
           defprops.put("test_duration", ""+win.test_duration);
           defprops.put("tcorr", ""+win.tcorr);
           defprops.put("trate", ""+win.trate);
        defprops.put("degree",""+win.degree);
           defprops.put("iterations", ""+win.iterations);
           defprops.put("shuffle","false");
```

```
    Properties props = new Properties(defprops);
    props.load(new FileInputStream("SendersTest.ini"));
    win.fullscreen = ((String) props.get("fullscreen")).equals("true")? true : false;
    win.logging = ((String) props.getProperty("logging")).equals("true")? true : false;
    win.verbose = ((String) props.getProperty("verbose")).equals("true")? true : false;
    win.test_duration = Integer.parseInt((String)props.getProperty("test_duration"));
    win.iterations = Integer.parseInt((String)props.getProperty("iterations"));
    win.tcorr = Double.parseDouble((String)props.getProperty("tcorr"));
    win.trate = Double.parseDouble((String)props.getProperty("trate"));
    int degree = Integer.parseInt((String)props.getProperty("degree"));
  if(degree<=win.degree) win.degree=degree;
    else System.out.println("ini file specifies degree exceeding "+win.degree);
    for(int g=0; g<win.g.length; g++)
        for(int i=0; i<win.degree; i++) {
      try {
            win.g[g].amp[i] =
              Double.parseDouble((String)
                            props.getProperty("g"+g+
                                        "amp"+i));
            win.g[g].phase[i] =
              Double.parseDouble((String)
                            props.getProperty("g"+g+
                                        "phase"+i));
        } catch(NullPointerException e) {
//              System.out.println("Error parsing parameters"
//                              +" in ini file ("+g+","+i+")");
      }
    }
  //System.out.println(defprops);
    //System.out.println(props);
    boolean shuffle =
        (String)props.getProperty("shuffle")=="true"?true:false;
    String tc_list = (String) props.getProperty("tcorr_list");
    if(tc_list!=null) {
        StringTokenizer str_tk = new StringTokenizer(tc_list," ");
        if(str_tk.countTokens()>0) {
          tcorr_list = new double[str_tk.countTokens()];
          String[] tok_list = new String[str_tk.countTokens()];
          for(int i=0; str_tk.hasMoreTokens(); i++)
              tok_list[i] = str_tk.nextToken();

          if(shuffle)
              Collections.shuffle(Arrays.asList(tok_list));

          Hashtable exid = new Hashtable();
          exid.put("A", new Double(0.25));
```

```
            exid.put("B", new Double(0.5));
            exid.put("C", new Double(0.75));
            exid.put("D", new Double(1.0));

            for(int i=0; i<tok_list.length; i++) {
                    if(exid.containsKey(tok_list[i]))
                        tcorr_list[i] =
                                ((Double)exid.get(tok_list[i])).doubleValue();
                    else try {
                        tcorr_list[i] = Double.parseDouble(tok_list[i]);
                    } catch(NumberFormatException e) {
                        tcorr_list[i] = 1.0;
                    }
            }
            System.out.print("tcorr_list: ");
            for(int i=0; i<tcorr_list.length; i++)
                    System.out.print(" "+tcorr_list[i]);
            System.out.println("");
        }
}
String sff_list = (String) props.getProperty("ff_list");
if(sff_list!=null) {
        StringTokenizer str_tk = new StringTokenizer(sff_list," ");
        if(str_tk.countTokens()>0) {
            ff_list = new String[str_tk.countTokens()];
            for(int i=0; str_tk.hasMoreTokens(); i++)
                    ff_list[i] = str_tk.nextToken()+".ser";

            System.out.print("ff_list: ");
            for(int i=0; i<ff_list.length; i++)
                    System.out.print(" "+ff_list[i]);
            System.out.println("");
        }
}
String svl_list = (String) props.getProperty("vl_list");
if(svl_list!=null) {
        StringTokenizer str_tk = new StringTokenizer(svl_list," ");
        if(str_tk.countTokens()>0) {
            vl_list = new String[str_tk.countTokens()];
            for(int i=0; str_tk.hasMoreTokens(); i++)
                    vl_list[i] = str_tk.nextToken();

            System.out.print("vl_list: ");
            for(int i=0; i<vl_list.length; i++)
                    System.out.print(" "+vl_list[i]);
            System.out.println("");
```

```
            }
        }
        String sbw_list = (String) props.getProperty("bw_list");
        if(sbw_list!=null) {
            StringTokenizer str_tk = new StringTokenizer(sbw_list," ");
            if(str_tk.countTokens()>0) {
                bw_list = new String[str_tk.countTokens()];
                for(int i=0; str_tk.hasMoreTokens(); i++)
                    bw_list[i] = str_tk.nextToken();

                System.out.print("bw_list: ");
                for(int i=0; i<bw_list.length; i++)
                    System.out.print(" "+bw_list[i]);
                System.out.println("");
            }
        }
        bcommand = (String) props.getProperty("bcommand");
        ecommand = (String) props.getProperty("ecommand");
    } catch(IOException e) { System.out.println("No ini file"); };

    if(argv.length>0)
        win.tcorr = Double.parseDouble(argv[0]);

    if(win.fullscreen)
        win.setSize(Toolkit.getDefaultToolkit().getScreenSize());

    if(tcorr_list==null) {
        win.fit();
        if(win.verbose) win.info();
    }
    win.show();

    // Retrieve or create the honor roll
    String hrfile = "HonorRoll.ser";
    HonorRoll roll;
    if (new File(hrfile).exists()) {
        try {
            FileInputStream fis = new
                FileInputStream(hrfile);
            ObjectInputStream ois = new
                ObjectInputStream(fis);
            roll = (HonorRoll)ois.readObject();
            fis.close();
        } catch(Exception e) {
            System.err.println(e);
            roll = new HonorRoll();
```

```
    }
} else {
roll = new HonorRoll();
}

// Main loop
int expn = 0;
while(true) {
    if(tcorr_list!=null || ff_list!=null
      || vl_list!=null || bw_list!=null) {
            if(tcorr_list!=null)
                win.tcorr = tcorr_list[expn % tcorr_list.length];
            if(ff_list!=null)
                win.ppfile = ff_list[expn % ff_list.length];
            if(vl_list!=null)
                win.set_val(vl_list[expn % vl_list.length]);
            if(bw_list!=null)
                win.set_bws(bw_list[expn % bw_list.length]);
            expn++;
            win.fit();
            if(win.verbose) win.info();
    }

    for(int i=0; i<win.g.length; i++) { // get all gauges to 0
            win.g[i].theta = 0;
            win.g[i].repaint();
    }

    if(!suppress_id) id.show();
    suppress_id = false;

    PrintStream logfile = null;
    if(logging) {
            try {
                logfile =
                        new PrintStream(new FileOutputStream(name+"."+expn+".log"));
                Date date = new Date();
                logfile.println("# ID "+name+" ("+date+")");
                win.info(logfile);
            } catch(FileNotFoundException e) {
                logfile = null;
                System.err.println(e);
            }
    }

    Process waitingProcess;
```

```
            if(win.bcommand!=null)
                try {
                    waitingProcess = Runtime.getRuntime().exec(bcommand);
                } catch(IOException e) { System.err.println(e); }

            wait(3);        // wait 3 sec to prepare

            win.tm.init(logfile);
            System.out.println("\007");

            wait(test_duration*60); // wait while test is performed

            win.tm.freeze();  // at the end freeze gauges for 1 sec
            wait(1);        // to allow latest alarms detection

            win.tm.finish();
            if(logging) logfile.close();
            if(verbose) System.out.println("");
            if(win.ecommand!=null)
                try {
                    waitingProcess = Runtime.getRuntime().exec(ecommand);
                } catch(IOException e) { System.err.println(e); }

            roll.add(name,win.tm.score);
            if(verbose) System.out.println("Top results:\n"+roll);
            try {
                FileOutputStream fos = new FileOutputStream(hrfile);
                ObjectOutputStream oos = new ObjectOutputStream(fos);
                oos.writeObject(roll);
                oos.close();
            } catch(Exception e) {
                System.err.println(e);
            }

            //st = new StatDlg(win);
            st = new ScoreDlg(win);
            st.show();
        }
    }
}
class Timer extends Thread {
    Gauge[] target;
    int interval;
    int t;
    int detected = 0;
    int alarm_id = -1;
```

```
double  award = 0;
double  score = 0;
int false_alarms = 0;
boolean running = false;
boolean frozen = false;
long time0, timeup;
boolean skip = false;

PrintStream logfile = null;

public Timer(Gauge[] g, int i) {
        target = g; interval = i;
        init();
        setDaemon(true);
}
public void init() {
        init(null);
}
public void init(PrintStream _logfile) {
        t = 0; score = 0; false_alarms = 0;
        logfile = _logfile;
        time0 = timeup = System.currentTimeMillis();
        running = true;
        frozen = false;
}
public void freeze() {
        frozen = true;
}
public void finish() {
        running = false;
}
public void run() {
        String noise = ""; // was "\007"
        while(true) {
            try { sleep(interval); }
            catch(InterruptedException e) {}
            if(!running) continue;
            if(!frozen) t = (int) (System.currentTimeMillis() - time0);

            // try to fix the frame rate to 20 fps
            int dt = (int) (System.currentTimeMillis() - timeup);
            int fps = (int) Math.floor(1000/(dt>0?dt:1));
            //final int dte = 1000/20;
            //if(dt<dte) { skip=true; continue; }
            //if(!skip) System.out.println("-- Slow CPU warning --");
            //skip = false;
```

```
            timeup = System.currentTimeMillis();

            NumberFormat f = new DecimalFormat(" 0.00;-0.00");
            NumberFormat fp = new DecimalFormat("+0.00;-0.00");
            NumberFormat fi = new DecimalFormat(" 00;-00");

            String prefix = ""+fi.format(fps)+" ";
            for(int i=0; i<target.length; i++)
                prefix += fi.format(target[i].latency)+" ";

            boolean false_alarm = detected>0;
            double  false_alarm_value = -1;

            int detected_alarm = -1;   // earliest alarm outstanding
            int detected_latency = -1; // how much time it took to detect it
            for(int i=0; i<target.length; i++) {
                double value = target[i].value;
                boolean out = Math.abs(target[i].theta) > target[i].limit;
                int latency = target[i].latency;
                boolean late = latency > target[i].maxlatency;

                if(late) {
                   target[i].latency = -1;        // missed alarm
                   target[i].amissed++;
                   double delta = -value;
                   score += delta;
                   System.out.println(prefix+"missed alarm   "+i+
                                       "\tscore "+f.format(score)+
                                      " ("+fp.format(delta)+")");
                   continue; // it fixes the bug with new accounting
                }

                if( latency>0 )
                   target[i].latency++;           // count up
                else if (latency==0 && out) {
                   target[i].atotal++;           // new alarm
                   target[i].latency++;
                }

//              if(detected>0 && latency>0) {
//                 if(detected_alarm == -1 ||
//                   target[detected_alarm].latency<target[i].latency)
//                      detected_alarm = i;
//              }

                if(detected>0 && latency>0 && i==alarm_id)
```

```
            detected_alarm = i;

          if( out == false && latency == -1)
            target[i].latency = 0;        // return to initial state
      }
      if(detected_alarm!=-1) {
          detected_latency = target[detected_alarm].latency;
          target[detected_alarm].latency = -1;
          target[detected_alarm].adetected++;
          double value = target[detected_alarm].value;
          double delta = value;
          score += delta;
          System.out.println(prefix+noise+
                      "detected alarm "+detected_alarm+
                      "/"+alarm_id+
                      "\tscore "+f.format(score)+
                      " ("+fp.format(delta)+")");
          false_alarm = false;
      }
      if(false_alarm) {
          false_alarms++;
          double delta =
            alarm_id>=0? -target[alarm_id].value : false_alarm_value;
          score += delta;
          System.out.println(prefix+"false alarm:    "+
                      "\tscore "+f.format(score)+" ("+
                      fp.format(delta)+")");
      }

      for(int i=0; i<target.length; i++)
          target[i].tick(t);

      if(logfile != null) {
          logfile.print(f.format(t/1000.)+" ");
          for(int i=0; i<target.length; i++)
            logfile.print(f.format(target[i].theta)+" ");
          logfile.print(detected>0?"D"+alarm_id+" "+
                     f.format(detected_latency*interval/1000.):"");
          for(int i=0; i<4; i++)
            if(target[i].latency>target[i].maxlatency)
                logfile.print(" M"+i);
          logfile.println("");
      }
      System.out.print(prefix+"score = "+f.format(score)+"\r");
      detected = 0;
  }
```

```
    }
}
interface Timed {
    public void tick(int t);
}
abstract class Gauge extends Canvas implements Timed, Cloneable {
//     public Signal sig;
    public int id;              // gauge number
    public static int degree = 3;      // the number of harmonics
    public double amp[];        // harmonics amplitude
    public double phase[];      // harmonics phases
    public double freq[];       // harmonics frequencies
    public double _ape = -1;    // unevaluated ape value
    public double mmgap = -1;   // min max gap during alarm

    public double bandwidth;
    public int size;
    public double value, t;
    public double theta = 0;    // initial angle
    public double limit = Math.PI*0.25; // theta limit
    public int latency = 0;     // count latency in detecting alarm
    public int maxlatency = 20; // defines when the alarm is missed

    public int atotal,adetected,amissed;

    public Gauge() {
//          sig = new Signal();
            amp = new double[degree];
            phase = new double[degree];
            freq = new double[degree];
            setparam(200,0.5,2.0);
            for(int i=0; i<degree; i++) {
                amp[i] = (1.2+0.5*(1-2*Math.random()))/degree;
                //phase[i] = Math.random()*2*Math.PI;
                phase[i] = 0;
                freq[i] = i==0? 1 : 0.5*freq[i-1];
            }
            init();
    }
    public Gauge(int size, double value, double bandwidth) {
            setparam(size,value,bandwidth);
//      sig = new Signal();
            amp = new double[degree];
            phase = new double[degree];
            freq = new double[degree];
            for(int i=0; i<degree; i++) {
```

```
            amp[i] = (1.2+0.5*(1-2*Math.random()))/degree;
            //phase[i] = Math.random()*2*Math.PI;
            phase[i] = 0;
            freq[i] = i==0? 1 : 0.5*freq[i-1];
        }
        init();
    }
    public void setparam(int size, double value, double bandwidth) {
        this.size = size;
        this.value = value;
        this.bandwidth = bandwidth;
        _ape = -1;
    }
    public void setparam(Signal sig) {
        degree = sig.amp.length;
        amp = sig.amp;
        phase = sig.phase;
        freq = sig.freq;
    }
    public void getparam(Signal sig) {
        sig.amp = amp;
        sig.phase = phase;
        sig.freq = freq;
    }
    public void init() {
        theta = 0;
        atotal = adetected = amissed = 0;
    }
    public Object clone() {
        try {
            Gauge g = (Gauge) super.clone();
            g.phase = (double[]) phase.clone();
        g.amp = (double[]) amp.clone();
        g.freq = (double[]) freq.clone();
            return g;
        } catch (CloneNotSupportedException e) {
            throw new InternalError(e.toString());
        }
    }
    public Dimension getPreferredSize() {
        return new Dimension(size+1,size+1);
    }
    public double f(int t) {
        double f = 0;
        double k = 0.001;
//      for(int i=0; i<degree; i++) {
```

```
//            f += amp[i]*Math.sin(phase[i]+k*bandwidth*t);
//            k *= 0.5;
//           //k -= 0.00025;
//        }
          // need to change it to
          for(int i=0; i<degree; i++)
            f += amp[i]*Math.sin(phase[i]+freq[i]*bandwidth*k*t);

          //f *= amp/degree;
          //theta = amp*(Math.sin(bandwidth*t*0.001)+
          //              Math.sin(0.5*bandwidth*t*0.001)+
          //              Math.sin(0.25*bandwidth*t*0.001));
          return f;
    }
    public double clip(double x, double min, double max) {
          return x < min ? min : x > max ? max : x;
    }
    public void tick(int t) {
          theta = f(t);
          repaint();
    }
    public double ape() {
          if(_ape > 0) return _ape;
          else return ape(10*60*1000,20,limit);
    }
    public double ape(int tries, int interval, double lim) {
          int alarms1 = 0; // alarm of the first kind (crossing the boundary)
          int alarms2 = 0; // alarm of the second kind (being out of range)
          boolean red = Math.abs(f(0))>lim;    // general case of f
          double minmaxgap = 1000, maxgap = 0;

          for(int t=1; t<=tries; t += interval) {
            double gap = Math.abs(f(t))-lim;
            if(gap>0) {
                  alarms2++;
                  if(!red) alarms1++;
                  red = true;
                  if(gap>maxgap) maxgap = gap;
            } else {
                  if(red) {
                    if(maxgap<minmaxgap) minmaxgap = maxgap;
                    //System.out.println("Gap "+maxgap);
                    maxgap=0;
                  }
                  red = false;
            }
```

```
        }
        mmgap = (minmaxgap<1000?minmaxgap:-1);
        double ape = (double) alarms1/tries*1000; // alarms per second
        _ape = ape;                             // for backward compatibility
        //System.out.println("MMGap "+id+": "+mmgap+" ape="+ape);
        return ape;
    }
    final boolean double_buffer = false;
    Image imgbuf = null;
    Graphics gc;
    public void update(Graphics g) {
        if(!double_buffer)
            paint(g);
        else {
            if(imgbuf==null) {
                imgbuf = createImage(getSize().width, getSize().height);
                gc = imgbuf.getGraphics();
            }
            paint(gc);
            g.drawImage(imgbuf, 0, 0, this);
        }
    }
}

class RoundGauge extends Gauge implements Timed {
//    public RoundGauge(int size, double value, double bandwidth) {
//        super(size,value,bandwidth);
//    }
    public void paint(Graphics g) {
        int arm = 0, crit = 150, val = 200, lim = 10;
        double th = clip(theta, -Math.PI/2, Math.PI/2);

      int x0 = (int) Math.round((getSize().width-size)/2.);
      int y0 = (int) Math.round((getSize().height-size)/2.);

        g.setColor(new Color(val,val,val));
        g.fillArc(x0,y0,size,size,45,90);
//      int lang = (int) Math.round(90-360/Math.PI*limit);
//      int rang = (int) Math.round(90+360/Math.PI*limit);
//      g.fillArc(x0,y0,size,size,lang,rang);

        g.setColor(new Color(crit,crit,crit));
        g.fillArc(x0,y0,size,size,45,-270);
        g.setColor(new Color(val,val,val));
        g.fillRect((int) Math.round(x0+size/2.-size*value/40),
              (int) Math.round(y0+size*3./4-size/20.),
```

```
                (int) Math.round(size*value/20),
                    (int) Math.round(size/10.));
        g.setColor(new Color(lim,lim,lim));
        g.drawRect((int) Math.round(x0+size/2.-size*value/40),
                (int) Math.round(y0+size*3./4-size/20.),
                (int) Math.round(size*value/20),
                    (int) Math.round(size/10.));
        g.setColor(new Color(arm,arm,arm));
        g.drawOval(x0,y0,size,size);

        int xc = (int) Math.floor(size/2), yc = (int) Math.floor(size/2);

        g.drawLine(x0+xc,y0+yc,
                    x0+xc+(int) Math.round(size/2*Math.sin(th)),
                    y0+yc+(int) Math.round(-size/2*Math.cos(th)));

        g.setColor(new Color(lim,lim,lim));
        g.drawLine(x0+xc,y0+yc,
                    x0+xc+(int) Math.round(size/2*Math.sin(limit)),
                    y0+yc+(int) Math.round(-size/2*Math.cos(limit)));
        g.drawLine(x0+xc,y0+yc,
                    x0+xc+(int) Math.round(size/2*Math.sin(-limit)),
                    y0+yc+(int) Math.round(-size/2*Math.cos(-limit)));
    }
}

class RectGauge extends Gauge implements Timed {
//    public RectGauge(int size, double value, double bandwidth) {
//        super(size,value,bandwidth);
//    }
    public void paint(Graphics g) {
        int arm = 0, crit = 150, val = 200;
        //Color c = g.getColor();
        g.setColor(new Color(crit,crit,crit));
        g.fillRect(0,(int)(0.25*size),(int)(0.25*size),(int)(0.5*size));
        g.fillRect((int)(0.75*size),(int)(0.25*size), (int)(0.25*size), (int)(0.5*size));

        g.setColor(new Color(255,255,255));
        g.fillRect((int)(0.25*size),(int)(0.25*size),(int)(0.5*size),(int)(0.5*size));

        g.setColor(new Color(val,val,val));
        g.fillRect((int) Math.floor(size/2.-size*value/4),
                (int) Math.floor(size*3./4-size/20.),
                (int) Math.floor(size*value/2),
                    (int) Math.floor(size/10.));
        g.setColor(new Color(arm,arm,arm));
```

```
        int xc = (int) Math.floor(size/2), yc = (int) Math.floor(size/2);

        g.drawLine(xc+(int) Math.floor(size/2*Math.sin(theta)),
                yc-(int)(0.1*size),
                xc+(int) Math.floor(size/2*Math.sin(theta)),
                yc+(int)(0.1*size));
    }
}

class InitDlg extends Dialog implements ActionListener {
    TextField tf;
    SendersTest p;
    InitDlg(SendersTest _p) {
        super(_p, "Initial Dialog", true); // later set mod=true
        p = _p;
        add("North", new Label("Please enter your Id"));

        tf = new TextField(40);
        add("Center",tf);

        Panel cp = new Panel();
        cp.setLayout(new FlowLayout());
        Button ok = new Button("Start");
        cp.add(ok);
        ok.addActionListener(this);
        Button quit = new Button("Quit");
        cp.add(quit);
        quit.addActionListener(this);
        add("South",cp);
        pack();

        Dimension dd = getSize();
        Dimension dp = p.getSize();
        int x = Math.max((dp.width-dd.width)/2, 0);
        int y = Math.max((dp.height-dd.height)/2, 0);
        setLocation(x,y);
    }
    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();
        if(cmd.equals("Start")) {
            hide();
            System.out.println("Input name: "+tf.getText());
            p.name = tf.getText();
            //dispose();
        } else if(cmd.equals("Quit")) {
            System.exit(0);
```

```
        }
    }
}

class StatDlg extends Dialog implements ActionListener {
    TextField tf;
    StatDlg(SendersTest p) {
        super(p, "Results", true); // later set mod=true
        Panel top = new Panel(new GridLayout(3,1));
        top.add(new Label("Test statistic for "+p.name));
        top.add(new Label("Score: "+p.tm.score));
        top.add(new Label("False alarms: "+p.tm.false_alarms));
        add("North", top);

        Panel info = new Panel(new GridLayout(2,2));
        for(int i=0; i<p.g.length; i++) {
            Panel gauge = new Panel(new GridLayout(4,2));
            gauge.add(new Label("Gauge "+i+":"));
            gauge.add(new Label(""));
            gauge.add(new Label("  detected:"));
            gauge.add(new Label(""+p.g[i].adetected));
            gauge.add(new Label("  missed:"));
            gauge.add(new Label(""+p.g[i].amissed));
            gauge.add(new Label("  TOTAL:"));
            gauge.add(new Label(""+p.g[i].atotal));
            info.add(gauge);
        }
        add("Center",info);

        Panel cp = new Panel();
        cp.setLayout(new FlowLayout());
        Button ok = new Button("Ok");
        cp.add(ok);
        ok.addActionListener(this);
        add("South",cp);
        pack();

        Dimension dd = getSize();
        Dimension dp = p.getSize();
        int x = Math.max((dp.width-dd.width)/2, 0);
        int y = Math.max((dp.height-dd.height)/2, 0);
        setLocation(x,y);
    }
    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();
        if(cmd.equals("Ok")) {
```

```
            hide();
            dispose();
        }
    }
}

class ScoreDlg extends Dialog implements ActionListener {
    TextField tf;
    ScoreDlg(SendersTest p) {
        super(p, "Results", true); // later set mod=true


        Label scorel = new Label("Your score: "+p.tm.score, Label.CENTER);
        //scorel.setFont(new Font ("TimesRoman", Font.BOLD, 20));
        add("Center", scorel);

        Panel cp = new Panel();
        cp.setLayout(new FlowLayout());
        Button ok = new Button("Ok");
        cp.add(ok);
        ok.addActionListener(this);
        add("South",cp);
        pack();

        Dimension dd = getSize();
        Dimension dp = p.getSize();
        int x = Math.max((dp.width-dd.width)/2, 0);
        int y = Math.max((dp.height-dd.height)/2, 0);
        setLocation(x,y);
    }
    public Insets insets () {
        return new Insets (50, 50, 50, 50);
    }
    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();
        if(cmd.equals("Ok")) {
            hide();
            dispose();
        }
    }
}

class HonorRoll implements Serializable {
    static final long serialVersionUID = 168332003083100000L;
    static final int size = 10;
    int index = -1;
```

```
class Result implements Serializable {
        public String name;
        public double score;
        public Date date;
        public Result(String name, double score) {
            this.name = name; this.score = score; date = new Date();
        }
        public String toString() {
            NumberFormat fp = new DecimalFormat(" 0.00;-0.00");
            return fp.format(score)+"\t"+name+"\t"+date;
        }
}
Result roll[];
public HonorRoll() {
        roll = new Result[size];
        for(int i=0; i<size; i++)
            roll[i] = null;
}
public int add(String name, double score) {
        Result res = new Result(name,score);

        index = -1;
        for(int i=0; i<size; i++) {
            if(roll[i] == null) {
                    roll[i] = res;
                    if(index<0) index = i;
                    break;
            } else
                    if(roll[i].score>=res.score) continue;
                    else {
                        if(index<0) index = i;

                        Result temp = roll[i];
                        roll[i] = res;
                        res = temp;
                    }
        }
        return index;
}
public String toString() {
        String s = "";
        for(int i=0; i<size; i++) {
            if(roll[i]==null) break;
            s += "["+(i+1)+"] "+roll[i].toString()+(index==i?" !!":"")+"\n";
        }
        return s;
```

```
    }
}

class Goal implements Serializable {
    static final long serialVersionUID = 168332003083100002L;
    int test_duration;
    double trate, tcorr;
    Date date;
    final double eps = 0.001;

    Goal(int test_duration, double trate, double tcorr) {
        this.test_duration = test_duration;
        this.trate = trate;
        this.tcorr = tcorr;
        this.date = new Date();
    }
    public String toString() {
        return "Goal("+test_duration+", "+trate+", "+tcorr+")";
    }
    public boolean equals(Object obj) {
        Goal g = (Goal) obj;
        return test_duration==g.test_duration
            && Math.abs(trate-g.trate)<eps
            && Math.abs(tcorr-g.tcorr)<eps;
    }
    public int hashCode() {
        return test_duration*100000
            +(int)Math.floor(tcorr*10000+10000+trate*100);
    }
}
class Signal implements Serializable {
    static final long serialVersionUID = 168332003083100003L;
    static int degree = 3;
    double rate, corr, error, bandwidth;
    double amp[], phase[], freq[];
    Date date;

    Signal() {
        amp = new double[degree];
        phase = new double[degree];
        freq = new double[degree];
        for(int i=0; i<degree; i++) {
            amp[i] = (1.2+0.5*(1-2*Math.random()))/degree;
            //phase[i] = Math.random()*2*Math.PI;
            phase[i] = 0;
            freq[i] = i==0? 1 : 0.5*freq[i-1];
```

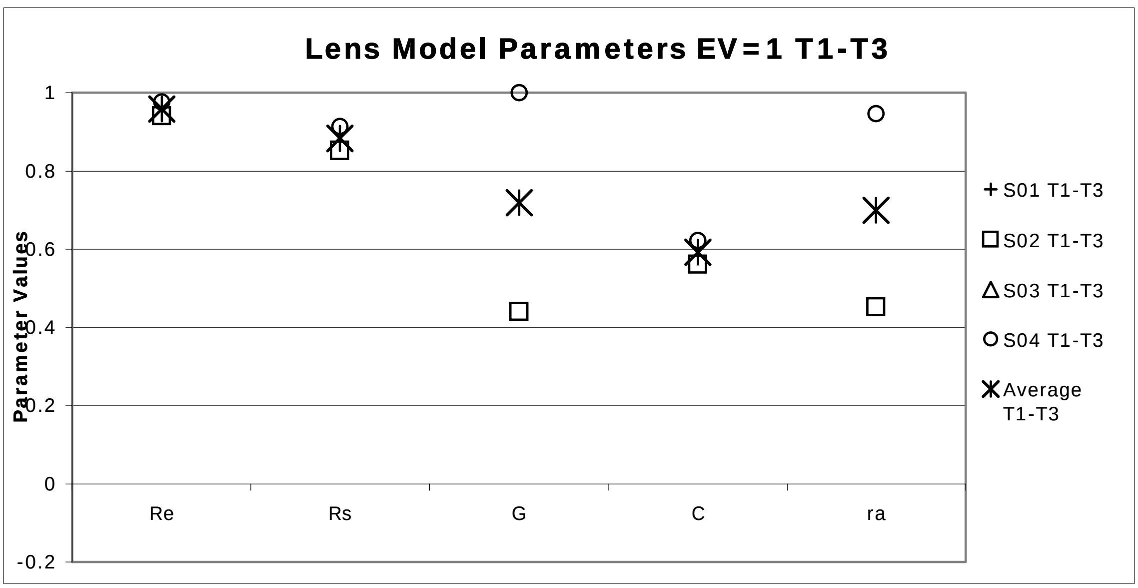
```
        }
        date = new Date();
    }
    Signal(double rate, double corr, double error,
             double amp[], double phase[], double freq[]) {
        this.rate = rate;
        this.corr = corr;
        this.error = error;
        this.amp = amp; // maybe cloning is needed here
        this.phase = phase;
        this.freq = freq;
        this.date = new Date();
    }
    Signal(Gauge g) {
        //rate = rate;
        //corr = corr;
        //error = error;
        amp = g.amp;
        phase = g.phase;
        freq = g.freq;
//      amp = (double[]) g.amp.clone(); // maybe cloning is needed here
//      phase = (double[]) g.phase.clone();
//      freq = (double[]) g.freq.clone();
        date = new Date();
    }
    public String toString() {
        NumberFormat fp = new DecimalFormat(" 0.00;-0.00");
        String s="";
//      s+= "# Sig corr/rate = ("+fp.format(corr)+","+fp.format(rate)+")\n#";
//      s+= "#";
//      for(int i=0; i<amp.length; i++) s+= "   Am"+i+"   Ph"+i; s+="\n";
        s+= "#";
        for(int i=0; i<amp.length; i++)
            s+=fp.format(amp[i])+" "+fp.format(phase[i])+" ";
        return s;
    }
}

class Population implements Serializable {
    static final long serialVersionUID = 168332003083100001L;
    Hashtable pop;
    int psize;
    Population() {
        pop = new Hashtable();
    }
    Signal[] get(Goal goal) {
```

```
        return (Signal[]) pop.get(goal);
    }
    void put(Goal goal, Signal[] sigs) {
        put(goal, sigs);
    }
    public String toString() {
        String s = "";
        for(Enumeration it = pop.keys(); it.hasMoreElements();) {
            Goal g = (Goal) it.nextElement();
            Signal sig[] = (Signal[]) pop.get(g);
            s+= g+":\n";
            for(int i=0; i<sig.length; i++) s+= ""+sig[i]+"\n";
        }
        return s;
    }
}
```

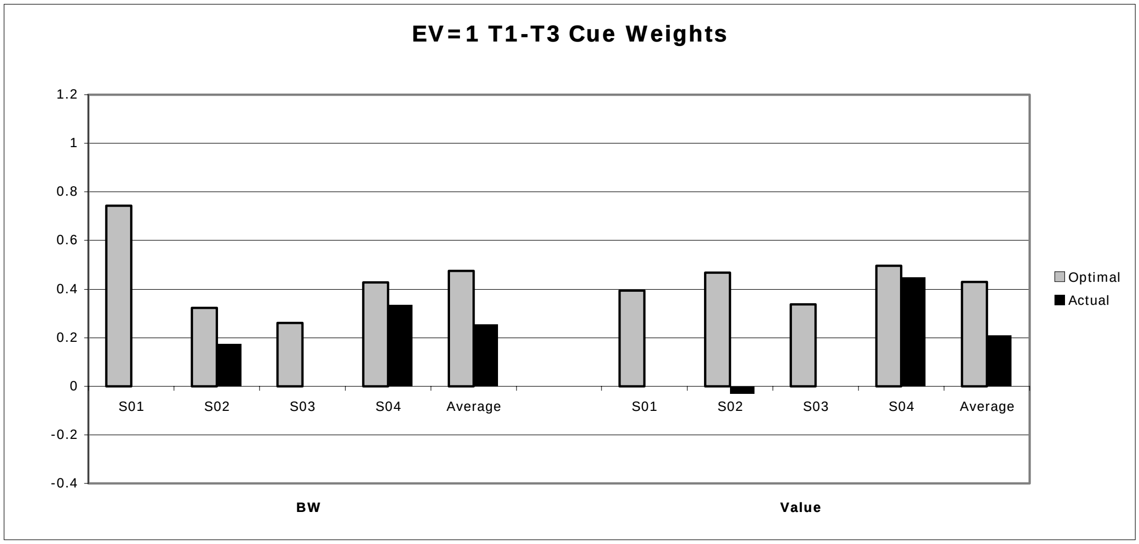
Appendix C

Lens Model Parameters (T1-T10)

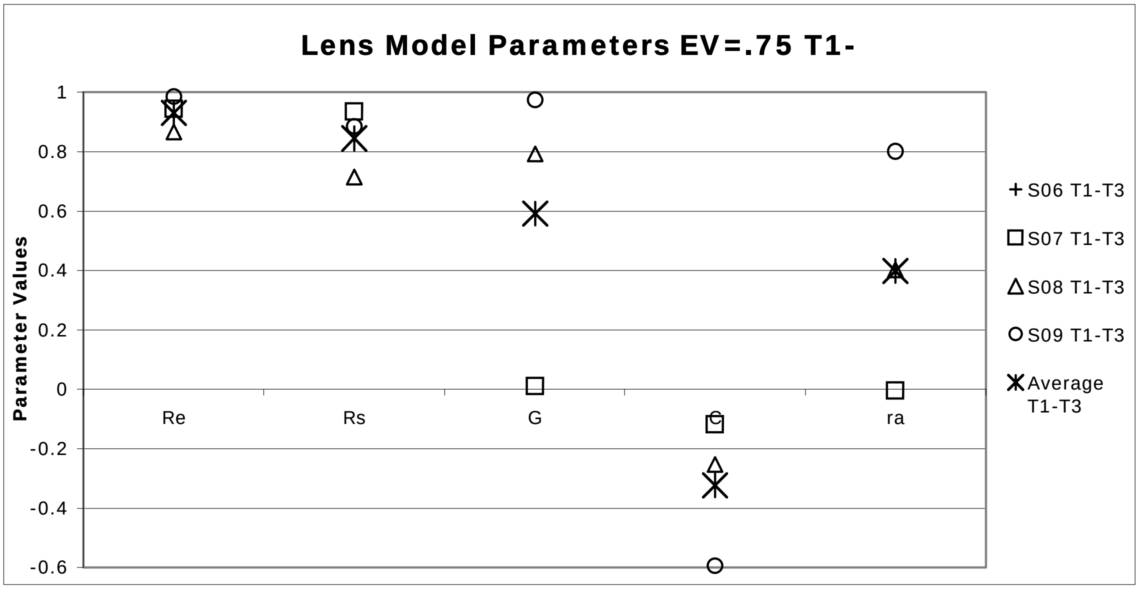
Trial 1 – Trial 3

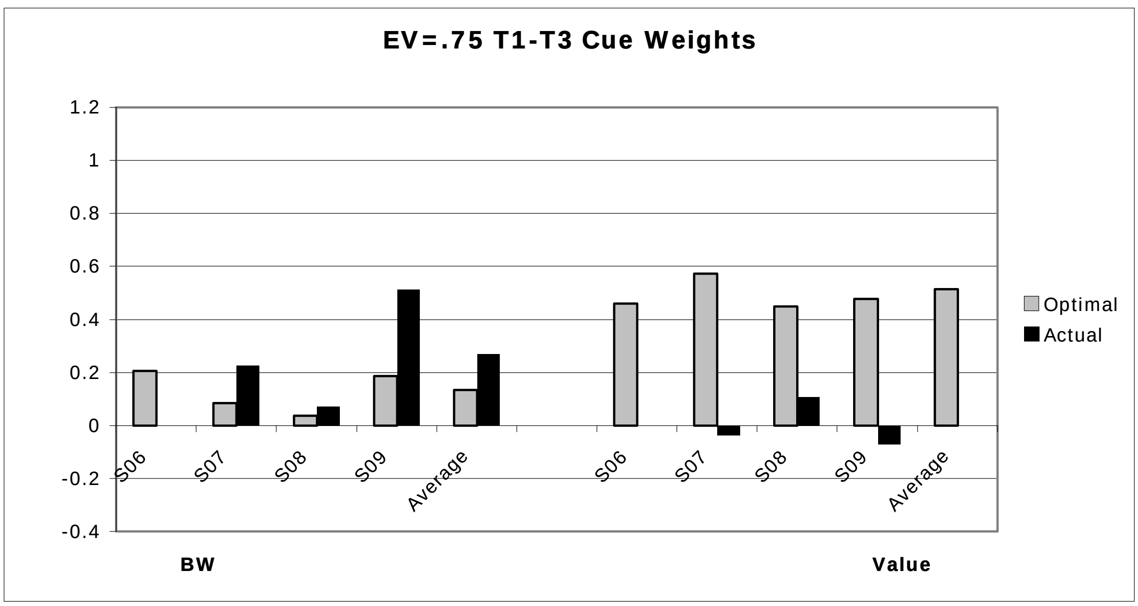


Trial 1 – Trial 3 continued

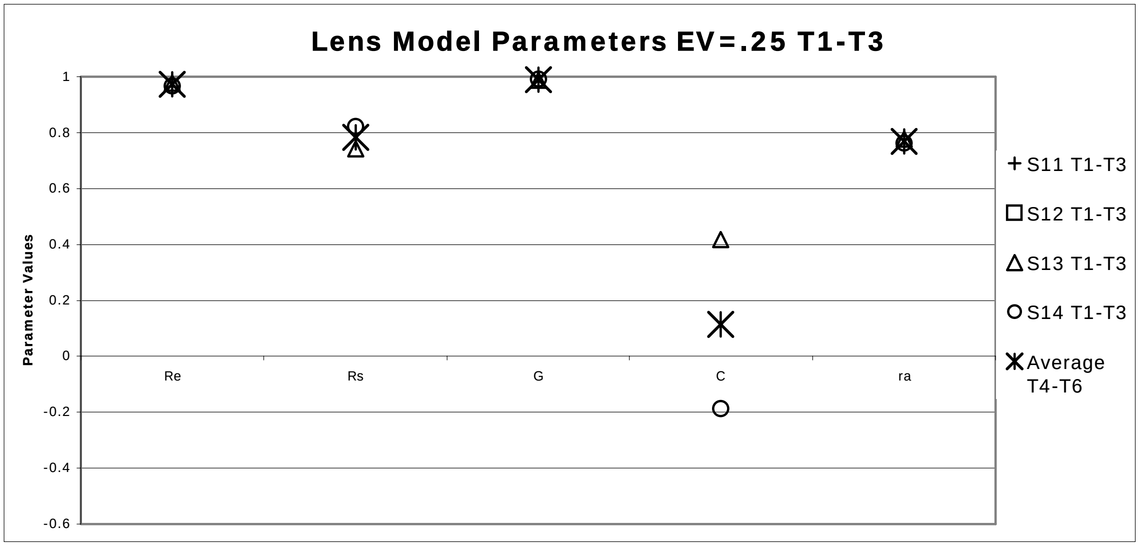


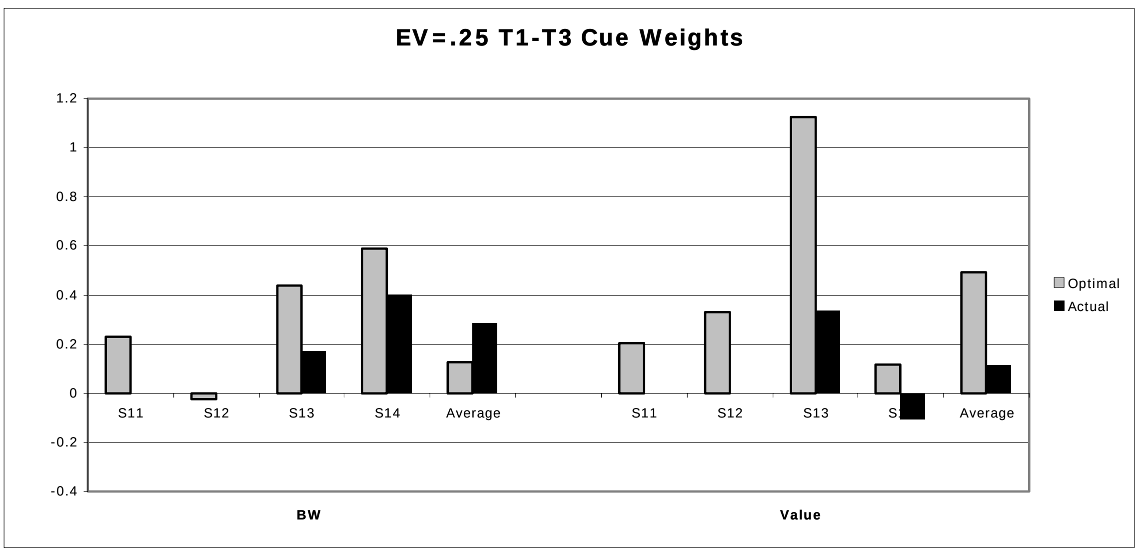
Trial 1 – Trial 3 continued



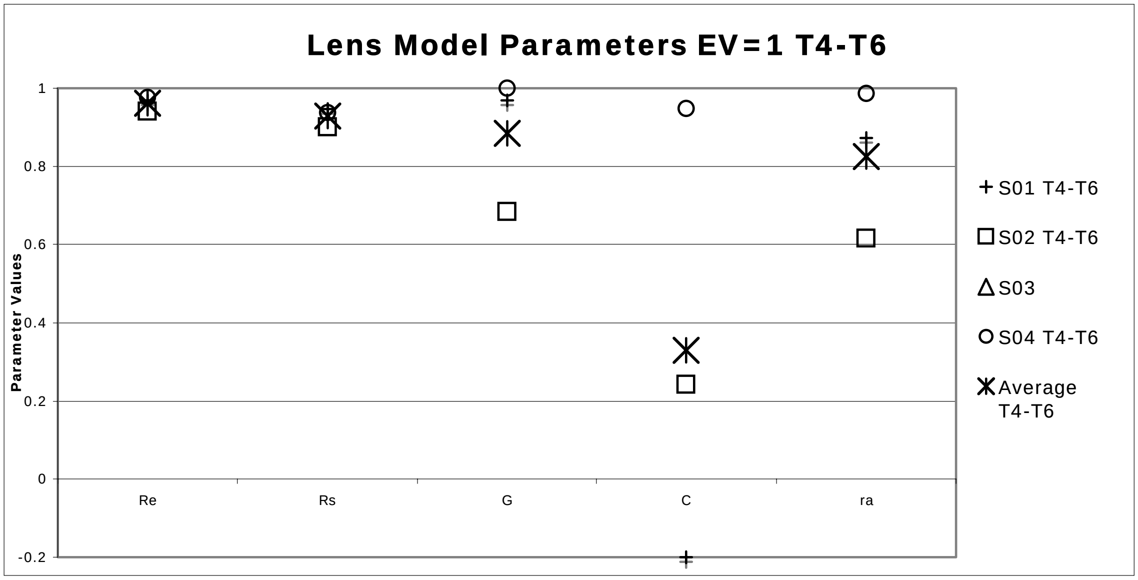


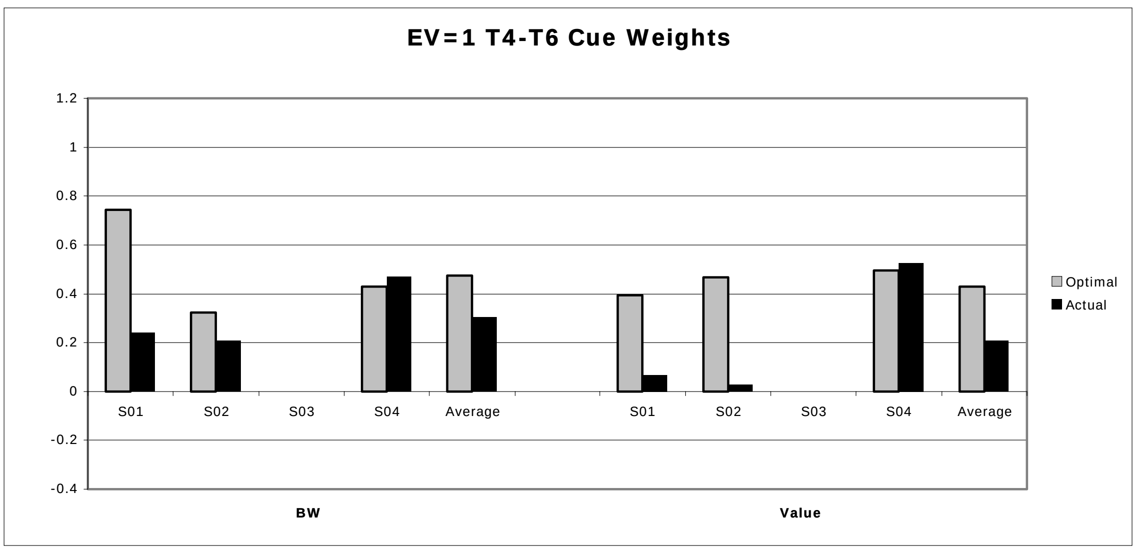
Trial 1 – Trial 3 continued



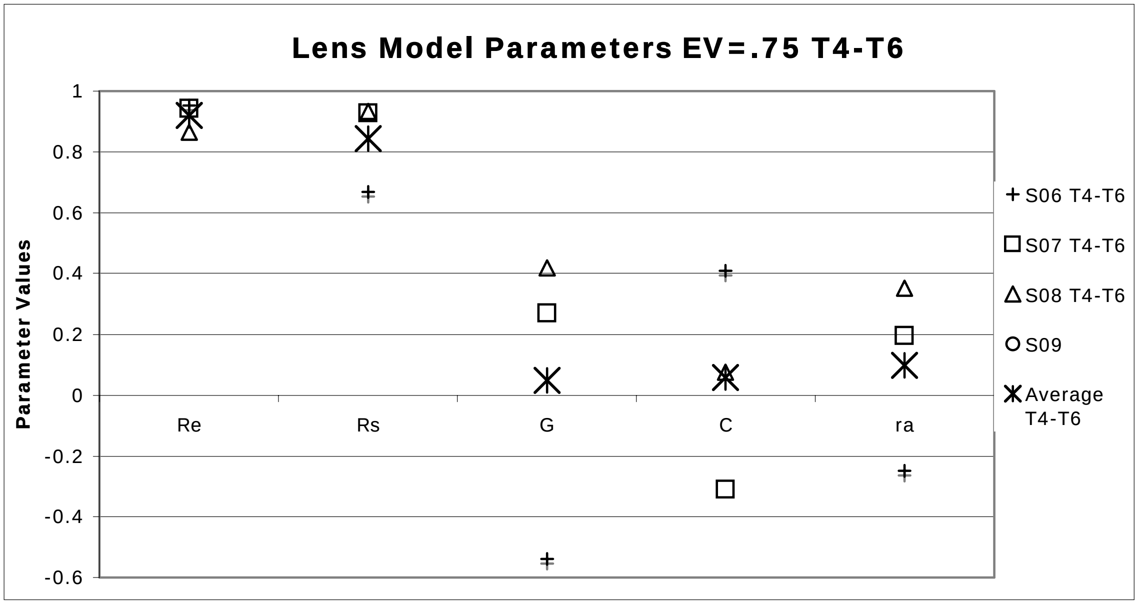


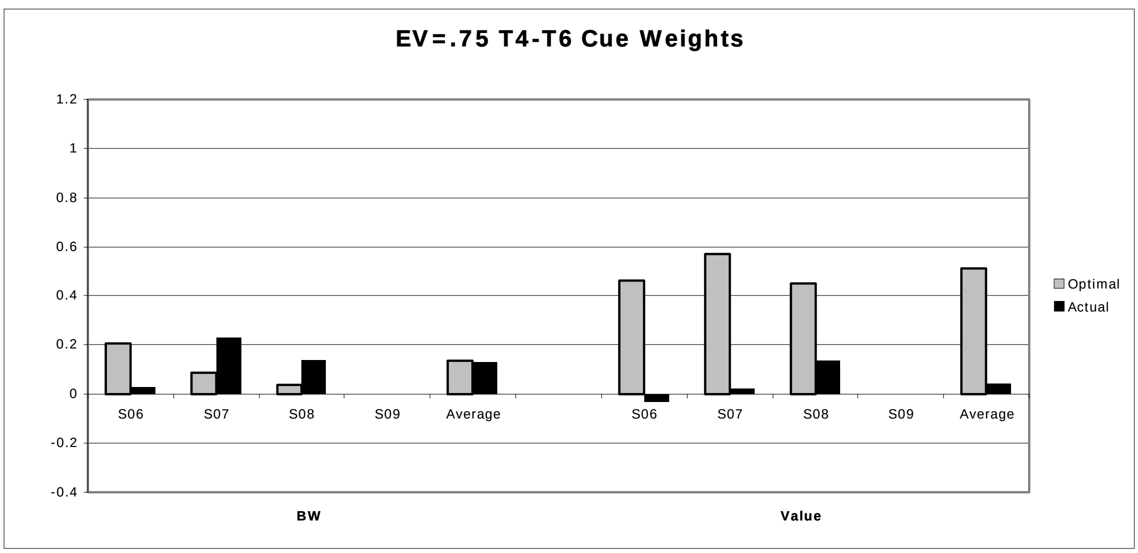
Trial 4 – Trial 6



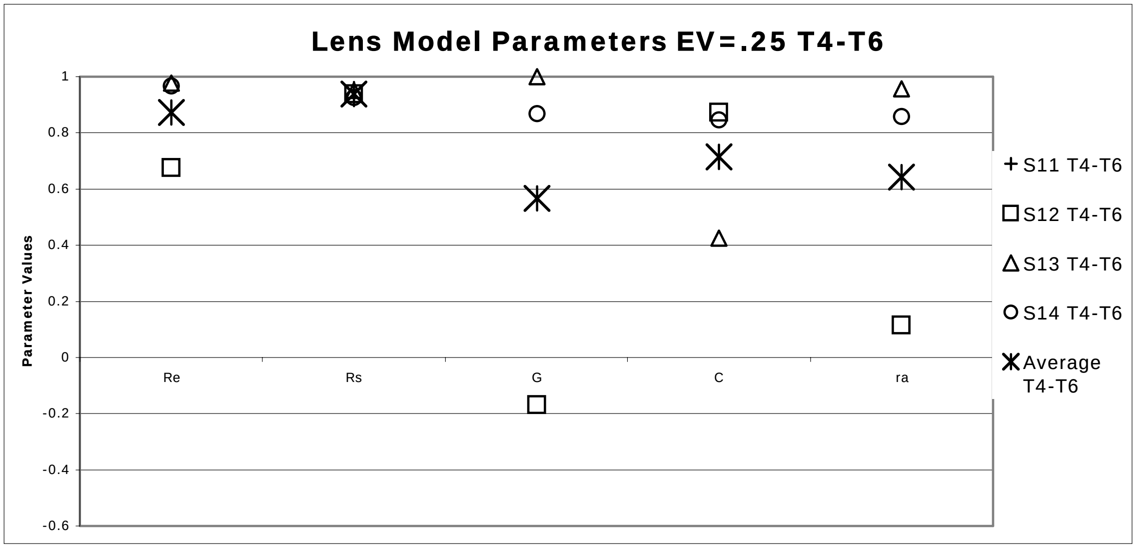


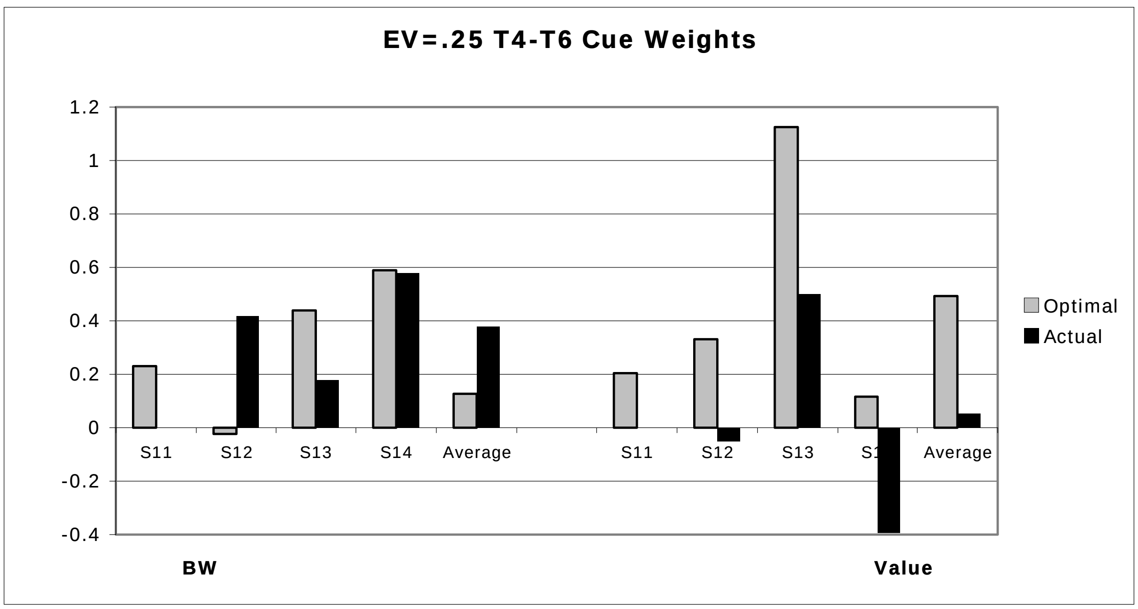
Trial 4 – Trial 6 continued



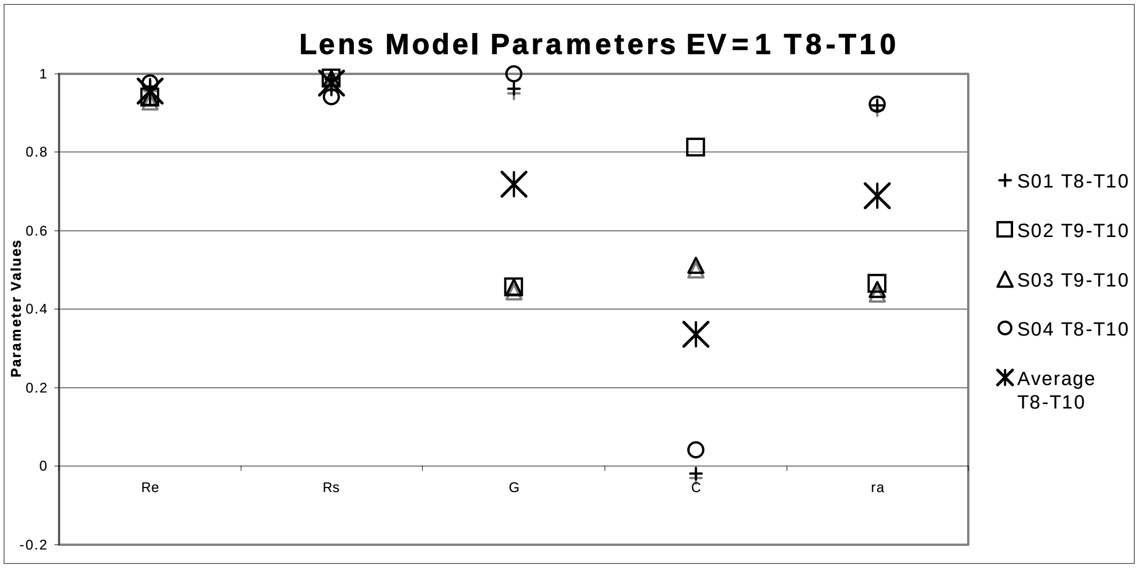


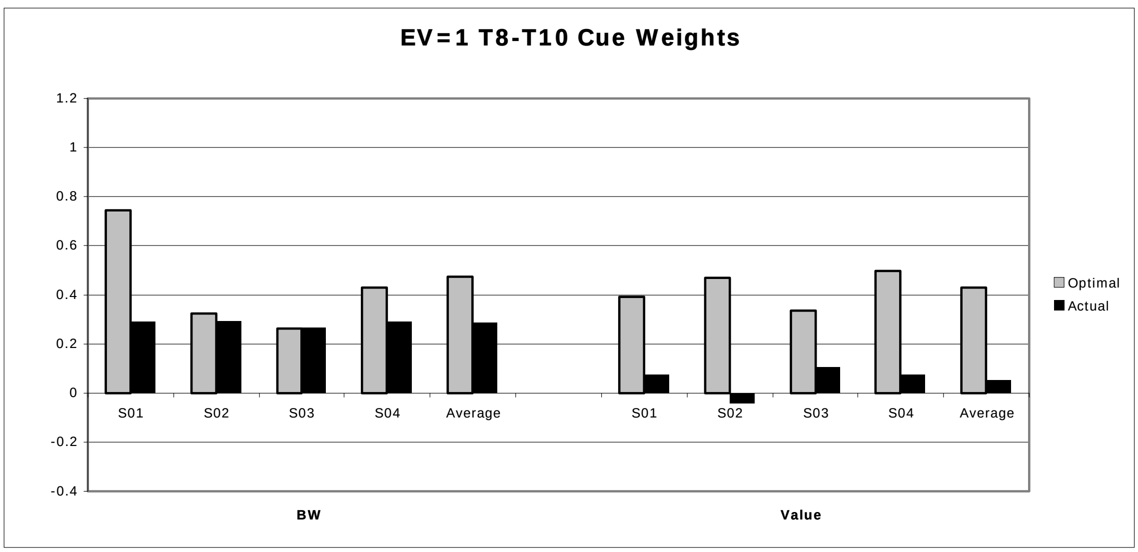
Trial 4 – Trial 6 continued



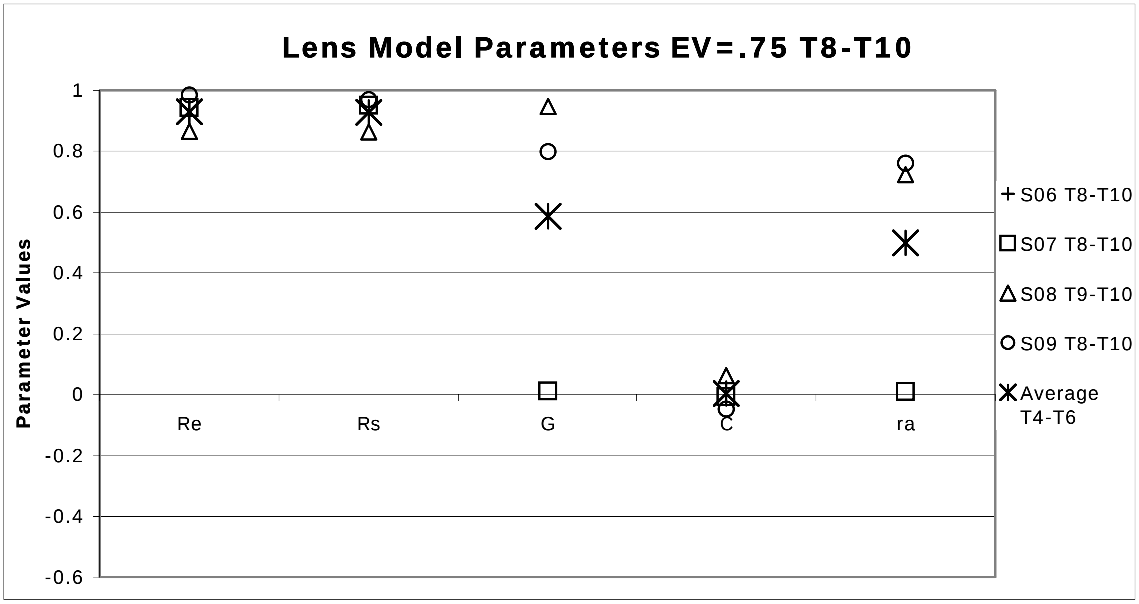


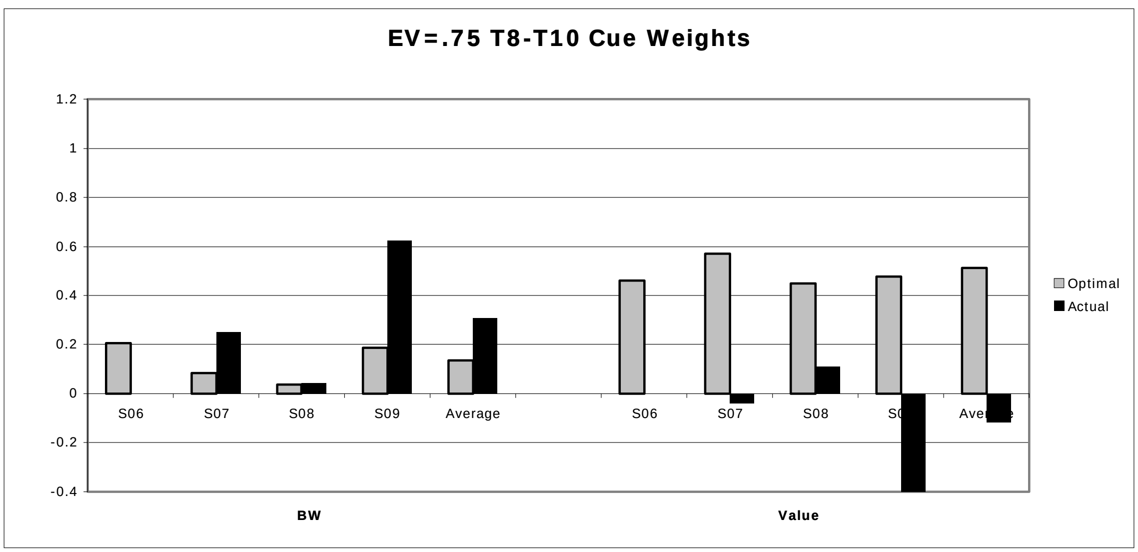
Trial 8 – Trial 10





Trial 8 – Trial 10 continued





Trial 8 – Trial 10 continued